Kasui Renku 1: Multimodal Expert Pruning, Mixed-Precision Sub-15 GB Quantization, and Coding Loss Localization for Qwen3.6-35B-A3B

Status: Technical Report (v1.0 — Released Control Artifact, Matched A100 Attribution, and Coding Localization Audit; Target-Hardware Fit and Release Benchmarks Pending).
Updated 24 November 2026. Primary training and attribution runtime: Google Colab Pro `NVIDIA A100-SXM4-80GB` (`167 GiB` host RAM) with pinned `llama.cpp @ 8a97fa98ab1a454709bbf5ac979661feaeadec40` (CUDA). Every reported metric is explicitly labeled `MEASURED`, `PROJECTED`, or `PENDING`, and cloud A100 CUDA memory measurements are kept strictly separate from unverified 16 GiB target-hardware claims. Authorship and institutional affiliations have not been assigned.

ABSTRACT

We document the design, compression pipeline, matched-hardware memory and throughput attribution, and stage-by-stage coding failure localization of **Kasui Renku 1**, a compressed multimodal Mixture-of-Experts (MoE) derivative of `Qwen/Qwen3.6-35B-A3B`. Starting from the 40-layer hybrid Gated DeltaNet + full-GQA multimodal backbone (`256` routed experts, `k = 8`), we apply 4-bucket multimodal REAP pruning (`256 → 168` routed experts, `-34.4%`, multiple of 8) while explicitly preserving 100% of vision (`visual.*`, `333` tensors) and single-head Multi-Token Prediction (`mtp.*`, `19` tensors) weights. We then heal the pruned checkpoint via pure-BF16 LoRA SFT targeting GQA, Gated DeltaNet, and `shared_expert` projections alongside all `40` unfrozen `mlp.gate` routers (`1,058` steps, final loss `0.416`, `90.12%` token accuracy) using verified `qwen/qwen4-max` traces, halting GRPO at step `10/150` after diagnosing prompt-key collisions (`300 → 140` prompts) and `72.5%` truncation. Exporting the clean SFT-only merge (`grpo_weight = 0.0`, inherited `mtp.*`) with a domain-matched importance matrix (`imatrix.dat`, `254.5 MB`) and mixed-precision APEX tensor protection yields the released control artifact `Kasui-Renku-1-APEX-Q4-sub15G.gguf` at **`14,879,344,192 B`** (`14.879 GB` decimal, `4.93 BPW`), paired with the mandatory vision projector `mmproj-Q8_0.gguf` (`610,655,168 B` / `0.611 GB`).

In 100%-matched `llama-server` attribution on `NVIDIA A100-SXM4-80GB` (`-np 1 -c 8192 -fa on -ngl 99`, speculation OFF) against official stock `Qwen3.6-35B-A3B-Q4_K_M.gguf` (`20.543 GB`, `256/k8`), `Kasui-Renku-1-APEX-Q4-sub15G` (`168/k6`) reduces peak VRAM by **`29.75%`** (`20,164 MiB → 14,166 MiB`), improves prompt prefill by **`+5.09%`** (`n_p = 793`) to **`+12.51%`** (`n_p = 3,133`), and lowers TTFT by **`-2.50%`** to **`-11.01%`**. Fixed-length decode throughput is `-3.43%` to `-3.83%` lower than Stock `Q4_K_M` (`123.94-125.07` vs. `128.34-130.05 tok/s`) because APEX's `Q8_0` dense/shared and `IQ4_XS` expert dequantization is ~7% heavier per active token at matched `k = 8`, partially offset by `k = 6`. On a matched 6-task repository bugfix suite (`repo_fix_01..06`, speculation OFF), Renku `168/k6` passed **`5/6`** at **`295.66`** **verified fixes/hour** (`2.00x` Stock `Q4_K_M`'s `4/6` and `147.97` fixes/hour). Enabling inherited `draft-mtp (n = 2)` accelerated decode to **`155.47 tok/s`** (`+24.20%` vs. `k = 6` nospec, `+19.42%` vs. Stock) at `54.52%` draft acceptance (`3,740/6,860`), but flipped `repo_fix_03` (`True → False`) via batch-verification floating-point tie divergence (`4/6`), so speculation remains disabled during quality attribution.

Paired diagnostics reveal important deficits and localize their root causes honestly: overriding routing to `k = 8` at inference regressed 26-case coding from **`7/26`** (`2/20` standalone `eval_code` + `5/6` `repo_fix`, `42.16` fixes/hr) to **`5/26`** (`3/20` + `2/6`, `33.27` fixes/hr, `-3.68%` decode speed), confirming **`k = 6`** as the control. Re-scoring all 52 `llama-cli` predictions with a hardened `</think>` + `entry_point`-aware AST extractor produced exact `0` recovery delta, establishing that all 19 `k = 6` failures are genuine model errors (`7` logic, `4` runtime/API, `5` `3,072`-token budget truncations, `2` syntax, `1` repetition loop). A six-stage one-variable-at-a-time localization ladder proves **zero**

> **quantization loss** between BF16/F16 and APEX Q4 (`2/20` `eval_code` on both BF16 Stage 4 and Q4 Stage 6; `1/6` identical on F16 vs. Q4 misses), exposes severe raw-pruning collapse (`0/26` on unhealed BF16 `pruned_168`), and documents strong sequence-cap sensitivity (`0/6` at `1,536` tokens vs. `4/26` at `3,072` tokens on healed BF16, with `9/20` `eval_code` items still hitting the `3,072` -token ceiling). Target-hardware 16 GiB Intel Arc 140V Vulkan verification, occupied 64K retrieval, sealed held-out vision, and the fixed ~12-task non-security Terminal-Bench 4.0 pilot remain `PENDING` following a verified `~66.32 GB` Google Drive archive handoff.

1. Introduction, Design Goals, and Epistemic Discipline

Deploying frontier-class multimodal coding assistants locally on thin-and-light laptops (such as a `32 GB` unified-RAM / `16 GiB` shared-VRAM Intel Arc 140V system) requires simultaneously reducing static weight footprint below `15.000 GB`, preserving native multimodal vision capability, maintaining low time-to-first-token (TTFT) and high verified-task throughput, and avoiding silent quality regressions from pruning or quantization.

Kasui Renku 1 targets this envelope by compressing `Qwen/Qwen3.6-35B-A3B` (`71.927 GB` BF16; `20.543 GB` official `Q4_K_M`) into a `≤15.000 GB` **decimal** mixed-precision GGUF paired with a **mandatory** vision projector (`mmproj-Q8_0.gguf` , `0.611 GB`). Text-only substitution is explicitly disallowed in the release specification.

Per our locked evaluation protocol (RENKU_EVALUATION_PROTOCOL.md), this technical report enforces four strict reporting rules:

1. **Explicit provenance labels:** Every number is labeled `MEASURED` , `PROJECTED` , or `PENDING` / `UNMEASURED` .
2. **Strict operating-regime separation:** Cloud `A100-SXM4-80GB` CUDA measurements (including `32K` and `64K` VRAM footprints) are never used to claim `16 GiB` local Intel Arc 140V Vulkan fit (`PENDING_TARGET_HARDWARE`).
3. **Diagnostic vs. held-out separation:** All five images in the `eval_50` smoke suite overlap `data/calib_manifest.json` ; accordingly, `eval_50` vision scores are reported strictly as Gate-0 diagnostics, never as sealed held-out vision generalization.
4. **Honest negative and null findings:** Raw pruning collapse, GRPO failure, standalone coding weakness (`2/20`), `k = 8` regression, per-token decode slowdown relative to `Q4_K_M` , and MTP tie-divergence flips are reported in full alongside wins.

---

2. Architecture and Compression Methodology

2.1 Base Architecture and Preserved Invariants

`Qwen/Qwen3.6-35B-A3B` (Apache-2.0) is a natively multimodal 40-layer hybrid Mixture-of-Experts model combining:

- **Hybrid attention:** 30 Gated DeltaNet linear-attention layers (F32 recurrent states) interleaved with 10 full Grouped-Query Attention (GQA) layers (every 4th layer, KV-cache quantized independently via `-ctk/-ctv`).
- **Sparse MoE FFN:** `256` routed experts (`top_k = 8` , `norm_topk_prob = True`) plus an always-active `shared_expert` .
- **Multimodal and speculative heads:** A native vision encoder/projector (`visual.*` , `333` tensors) and a single-head Multi-Token Prediction module (`mtp.*` , `19` tensors, mapped to `blk.40` in `llama.cpp`), nested under

`config.json → text_config`.

2.2 Phase 2: 4-Bucket Multimodal REAP Pruning (`256 → 168`)

In `prune_and_heal.py`, routed experts are pruned from `256` to **`168`** (`-34.4%`, exact multiple of 8 for GGML/Triton alignment; `checkpoints/pruned_168` = `49.19 GB` BF16, `~24.5B` total parameters):

- **Nested config synchronization:** `update_nested_qwen36_config` updates `text_config.num_experts = 168`, `text_config.num_experts_per_tok = 6`, and `text_config.norm_topk_prob = True` (plus top-level mirrors if present).
- **Mandatory tensor preservation:** Unlike standard REAP forks that drop auxiliary heads during checkpoint export, `prune_and_heal.py` explicitly copies all `visual.*` (`333/333`) and `mtp.*` (`19/19`) tensors into the pruned safetensors shards and `model.safetensors.index.json`, verified programmatically via `verify_preserved_tensors`.
- **4-bucket calibration:** Calibration drew from four buckets (`coding_swe_terminal`, `deep_reasoning_longctx`, `structured_general_abstention`, and `vision_screenshot_qa`, 0% Japanese). Post-hoc auditing (Section 4.3) identified two limitations of the v1 REAP pass: prompt-only calibration from `seed_prompts.jsonl` (without assistant code completions) and unmasked softmax dilution across inactive tail experts. Both have been fixed offline in `prune_and_heal.py` for any future v2 pruning pass.

2.3 Phase 3: Pure-BF16 Hybrid LoRA SFT, Router Healing, and GRPO Halt

Training data (`data/train.jsonl`) were distilled from `qwen/qwen4-max` via OpenRouter pinned strictly to Alibaba (`qwen/qwen4-max-20261115`, `allow_fallbacks: false`, zero plugins) under dual-mode routing (~81% `deep_think` at `effort = "high"` across coding, SWE, DevOps, GPQA STEM, and retrieval; ~19% `direct` at `effort = "none"` across strict JSON, abstention, and versatile Q&A). Cumulative OpenRouter spend across data generation was `$45.5019 / $50.00` (`$4.4981` remaining; `$0.00` spent during the audit and localization phase).

- **Phase 3A — Pure-BF16 LoRA SFT + Unfrozen Routers (`COMPLETE`, `MEASURED`):**
  - *Targeted LoRA projections:* GQA (`q_proj`, `k_proj`, `v_proj`, `o_proj`), Gated DeltaNet (`in_proj_qkv`, `in_proj_z`, `in_proj_a`, `in_proj_b`, `in_proj_qkvz`, `in_proj_ba`, `out_proj`), and `shared_expert` (`gate_proj`, `up_proj`, `down_proj`). Individual routed experts (`mlp.experts.*`) were excluded from LoRA; `visual.*` remained frozen; `neftune_noise_alpha = 5.0` was enabled.
  - *Full-weight router healing:* All **`40`** `mlp.gate` router modules (~10M parameters) were unfrozen directly so the softmax router could adapt to the surviving `168` experts under `k = 6`.
  - *Outcome:* Completed `1,058 / 1,058` steps (`checkpoints/sft/`, `40/40` healed `mlp.gate` tensors), reaching final training loss **`0.416`** and **`90.12%`** token accuracy.
- **Phase 3B — Verifiable GRPO (`HALTED AT STEP 10/150`, `MEASURED`):**
  - Live telemetry at step `10/150` (`52m 20s` elapsed, `310.3 s/step`, `~12.06 h` projected remaining against a ~20 h A100 budget) showed flat mean reward (`0.175` from step 5), **`72.5%`** completion truncation at `1,024` tokens, `frac_reward_zero_std = 0.50` (`0.80` at step 5), and near-zero `grad_norm = 0.0001281`.
  - Code forensics uncovered a prompt-key dictionary collision (`spec_by_prompt` collapsed `300` rows into `140` distinct prompt strings, causing `29` groups covering `189` rows to overwrite distinct `test_code` verifiers) plus

an unmatched `</think>` parser bug. GRPO was halted immediately to prevent wasted compute and model degradation.

- **Phase 3C — Clean SFT-Only Merge (`COMPLETE`, `MEASURED`):**
  - Merged with `grpo_weight = 0.0` into `checkpoints/renku_v1_sft_inherited_mtp` (`49.212 GB` across `26` safetensor shards, `updated = 693` tensors, `visual = 333`, `mtp = 19`), retaining the stock inherited single-head `mtp.*` weights without risking backbone drift.

2.4 Phase 4: Mixed-Precision APEX Q4 GGUF and Mandatory Vision Export

Using pinned `llama.cpp @ 8a97fa98ab1a454709bbf5ac979661feaeadec40` (CUDA) via `export_apex_gguf.sh`:

1. Converted `checkpoints/renku_v1_sft_inherited_mtp` to `Kasui-Renku-1-F16.gguf` (`48,337,864,384 B` / `48.338 GB`, preserving `blk.40` MTP tensors) and exported the mandatory multimodal projector `mmproj-F16.gguf` (`899,283,008 B` / `0.899 GB`) quantized to `mmproj-Q8_0.gguf` (`610,655,168 B` / `0.611 GB`).
2. Computed a domain-matched importance matrix (`imatrix.dat`, `254,539,584 B`) from `data/imatrix_calib.txt` (verified `qwen4-max` coding, `<think>`, and JSON tool traces; `57,926` tokens, ~28 chunks of 2,048).
3. Applied mixed-precision APEX tensor protection (`apex_tensor_types.txt`) with fail-fast `--dry-run` validation. Because keeping `blk.40` MTP routed experts at `Q8_0` produced `Kasui-Renku-1-APEX-Q4.gguf` at `15,132,575,296 B` (`15.133 GB`, `132.6 MB` over the `15.000 GB` decimal target), the exporter automatically generated the sub-15 GB sibling `Kasui-Renku-1-APEX-Q4-sub15G.gguf` (`14,879,344,192 B` / **`14.879 GB`**, **`4.93 BPW`**) by quantizing only `blk.40` MTP experts to `Q5_K` / `IQ4_XS` while keeping **100% of backbone layers `0..39` untouched**.

Table 1: APEX Mixed-Precision Tensor Quantization Recipe (`Kasui-Renku-1-APEX-Q4-sub15G.gguf`)

Tensor Pattern	Quantization Type	Architectural Role
`^token_embd\.weight$`, `^output\.weight$`	`Q8_0`	Input embeddings and LM output head (anchored to avoid matching `attn_output`)
`.*ffn_gate_inp.*`, `.*conv1d.*`	`F16`	MoE router gates and Gated DeltaNet 1D convolutions
`.*ssm_.*`, `.*ffn_.*_shexp.*`	`Q8_0`	Gated DeltaNet state-space weights and always-active shared experts
`.*nextn.*`, `blk\.40\.attn_.*`	`Q8_0`	Inherited MTP next-token projections and `blk.40` attention
`blk\.([0-9]|[1-3][0-9])\.ffn_down_exps\.weight`	`Q5_K`	Backbone layers `0..39` routed expert down-projections
`blk\.([0-9]|[1-3][0-9])\.ffn_(gate|up|gate_up)_exps\.weight`	`IQ4_XS`	Backbone layers `0..39` routed expert gate/up projections (imatrix-guided)
`blk\.40\.ffn_down_exps\.weight` / `blk\.40\.ffn_(gate|up|gate_up)_exps\.weight`	`Q5_K` / `IQ4_XS`	Sub-15 GB sibling override on MTP block `40` experts only (default `Q8_0` in `15.133 GB` build)

Table 2: Released Control Artifacts, Checksums, and Durable Archive Status (`MEASURED`)

Artifact	Exact Bytes	Size (GB dec)	SHA-256 Prefix	Role & Archive Status
`Kasui-Renku-1-APEX-Q4-sub15G.gguf`	`14,879,344,192`	`14.879 GB`	`bc4a60a15840..`	**Primary released control (`168`, `k=6`, `4.93 BPW`)** — Drive verified
`mmproj-Q8_0.gguf`	`610,655,168`	`0.611 GB`	`779a83c8d61a..`	**Mandatory vision projector (release default)** — Drive verified
`mmproj-F16.gguf`	`899,283,008`	`0.899 GB`	`870e04965f85..`	Reversible F16 vision projector source — Drive verified
`imatrix.dat`	`254,539,584`	`0.255 GB`	`7292fc39d732..`	Domain-matched importance matrix — Drive verified
`checkpoints/renku_v1_sft_inherited_mtp/`	`~49.212 GB`	`49.212 GB`	`26/26` shards	Full healed BF16 checkpoint — Drive verified (`errors = 0`)
`Kasui-Renku-1-APEX-Q4.gguf`	`15,132,575,296`	`15.133 GB`	`a026f8125ce8..`	Sibling build (`blk.40` experts at `Q8_0`) — reproducible from BF16 + imatrix
`Kasui-Renku-1-F16.gguf`	`48,337,864,384`	`48.338 GB`	—	Intermediate F16 GGUF (exceeded `18 GiB` `drivefs` cache) — reproducible

Initial fail-fast Phase 4 smoke checks on A100-80GB passed both text generation (`llama-cli`: clean `def add(a, b): return a + b`, `113.6 tok/s` prompt / `112.8 tok/s` decode) and multimodal chart reading (`llama-mtmd-cli` + `mmproj-Q8_0.gguf` on `img_01_bar_chart.png`: exact title and Q1 `$120M`, Q2 `$185M`, Q3 `$140M`, Q4 `$230M` in `121 ms` encode).

---

3. Matched A100-80GB Memory, Throughput, and Speculation Attribution (`MEASURED`)

3.1 Matched Experimental Protocol

All head-to-head speed, VRAM, and speculation measurements were executed on the exact same `NVIDIA A100-SXM4-80GB` using `llama-server` built from pinned `llama.cpp @ 8a97fa98ab1a454709bbf5ac979661feaeadec40` with identical serving flags (`-np 1 -c 8192 -fa on -ngl 99 -t 4 --no-warmup`, `cache_prompt = false`, `temperature = 0`, `seed = 42`, CPU teacher OFF).

To establish an unpruned Q4 control without touching Colab overlay disk (`18.954 GiB` free), official `ggml-org/Qwen3.6-35B-A3B-Q4_K_M.gguf` (`20,542,732,576 B`, `~4.74 BPW`, `256` experts, `k = 8`, `sha256 = f4ffe03c…` — closest official Q4 BPW match to Renku's `4.93 BPW`) was staged exclusively in RAM-backed `/dev/shm` (`82.175 GiB` free, `0` overlay disk bytes used, `0` Renku files deleted) and removed immediately after benchmarking. Third-party `Fable-5-Distill` GGUFs (`IQ3_M` / `IQ4_XS`) were explicitly excluded as mismatched controls.

3.2 Fixed-Length VRAM, Prefill, TTFT, and Decode Attribution (Speculation OFF)

Table 3 reports fixed-length benchmarks (`ignore_eos = true`, `n_g = 128`, 3 repeats per cell, speculation OFF).

Model / Effective k	File Size (Δ vs. Stock)	Peak VRAM (Δ vs. Stock)	n_p=793 TTFT	n_p=793 Prefill	n_p=793 Decode	n_p=3133 TTFT	n_p=3133 Prefill	n_p=3133 Decode
Stock Qwen3.6-35B-A3B-Q4_K_M (256/k8)	20.543 GB (ref)	20,164 MiB (ref)	327.5 ms	2,613.63 t/s	130.05 t/s	766.0 ms	4,125.57 t/s	128.34 t/s
Kasui-Renku-1-APEX-Q4-sub15G (168/k8)	14.879 GB (-27.57%)	14,190 MiB (-29.63%)	326.3 ms (-0.37%)	2,665.44 t/s (+1.98%)	120.56 t/s (-7.30%)	723.7 ms (-5.52%)	4,368.24 t/s (+5.88%)	119.53 t/s (-6.86%)
Kasui-Renku-1-APEX-Q4-sub15G (168/k6)	14.879 GB (-27.57%)	**14,166 MiB (-29.75%)**	**319.3 ms (-2.50%)**	**2,746.58 t/s (+5.09%)**	125.07 t/s (-3.83%)	**681.7 ms (-11.01%)**	**4,641.80 t/s (+12.51%)**	123.94 t/s (-3.43%)

Mechanistic attribution:

- **VRAM reduction (`-5,998 MiB` / `-29.75%`):** Pruning `88` routed experts per layer (`256 → 168`) plus APEX quantization lowers A100 peak VRAM from `20,164 MiB` to `14,166 MiB` (`14,190 MiB` at `k = 8`).
- **Prefill & TTFT gains:** At `n_p = 3,133`, reducing active experts to `k = 6` alongside `168` total experts yields `+12.51%` higher prefill throughput (`4,641.80` vs. `4,125.57 tok/s`) and `-11.01%` lower TTFT (`681.7` vs. `766.0 ms`).
- **Decode trade-off:** At matched `k = 8`, APEX is `-6.86%` to `-7.30%` slower per decoded token than Stock `Q4_K_M` because `Q8_0` dense/shared layers and `IQ4_XS` non-linear dequantization carry higher per-token memory/compute cost than uniform `Q4_K_M`. Reducing active experts to `k = 6` recovers roughly half of this gap (`-3.43%` to `-3.83%` vs. Stock `Q4_K_M`).

3.3 Matched 6-Task `repo_fix` End-to-End Throughput and Speculation Sweep

Table 4 reports matched `llama-server` (`-np 1`) runs across the 6-task repository bugfix suite (`repo_fix_01..06`).

Arm	Speculation	Pass Rate	Wall Time	Mean TTFT	Prefill (t/s)	Decode (t/s)	Draft Acceptance	Verified Fixes/Hr	Peak VRAM
stock_256_k8_nospec	OFF	4/6	97.32 s	244.3 ms	1,129.82	130.19	—	147.97	20,166 MiB
k6_nospec (Renku Control)	**OFF**	**5/6**	60.88 s	230.4 ms	1,192.69	125.18	—	**295.66 (2.00x)**	**14,166 MiB**
k8_nospec (Renku k=8)	OFF	2/6	56.33 s	234.5 ms	1,154.30	120.58	—	127.82	14,190 MiB
k6_draft_mtp_n2 (Inherited MTP)	draft-mtp (n=2)	4/6*	**48.32 s**	244.8 ms	1,135.64	**155.47 (+24.2%)**	**54.52%** (3,740/6,860)	**298.01**	14,768 MiB
k6_ngram_mod	ngram-mod	4/6	54.42 s	229.0 ms	1,194.39	123.71	16.88% (108/640)	264.61	14,166 MiB

**Speculation caveat:* Inherited `draft-mtp (n = 2)` increased decode speed by `+24.20%` over `k6_nospec` (`+19.42%` over Stock `Q4_K_M`) at `+602 MiB` VRAM overhead, confirming that preserved stock `mtp.*` weights remain strongly aligned with the pruned-and-healed backbone (`54.52%` acceptance). However, `draft-mtp` flipped `repo_fix_03` from `True` to `False` due to batch-verification floating-point tie divergence under greedy decoding. Speculation therefore stays OFF during all quality attribution runs and must pass deterministic stability verification before default enablement.

3.4 Separate `32K` and `64K` Context + Mandatory Vision Footprint (`A100-SXM4-80GB CUDA ONLY`)

Table 5 documents `Kasui-Renku-1-APEX-Q4-sub15G.gguf` + mandatory `mmproj-Q8_0.gguf` with `-ctk q8_0 -ctv q8_0 -fa on` on **`NVIDIA A100-SXM4-80GB` (CUDA)**. Because `30` of `40` layers use constant-memory Gated DeltaNet recurrent states and only `10` layers allocate full-attention KV cache, doubling the configured context window from `32K` to `64K` added only **`+436 MiB`** of VRAM.

Configured Context (`-c`)	Loaded VRAM	Peak VRAM (w/ `eval_vis_03`)	Host RAM Free	`eval_vis_03` Invoice OCR	TTFT	Prefill / Decode
`32,768` (`q8_0` KV, `-fa on`)	`14,952 MiB` (`14.60 GiB`)	**`15,062 MiB`** (`14.71 GiB`)	`165,116 MiB`	**PASS**	`473.4 ms`	`838.9 / 119.0 t/s`
`65,536` (`q8_0` KV, `-fa on`)	`15,388 MiB` (`15.03 GiB`)	**`15,498 MiB`** (`15.13 GiB`)	`165,061 MiB`	**PASS**	`467.9 ms`	`848.4 / 120.4 t/s`

Strict Regime Guardrails:

1. These are **CUDA A100-80GB** measurements on short image prompts (`eval_vis_03`), NOT occupied 64K retrieval tests and NOT Intel Arc 140V Vulkan measurements. Actual 16 GiB Intel Arc 140V Vulkan fit for `32K + vision` remains **`PENDING_TARGET_HARDWARE`**.
2. For reference context only, an earlier non-Renku local Intel Arc 140V Vulkan (`b11382`) hardware control (`15.81 GB` third-party `qwen35moe` GGUF, no speculation) measured `67.15 tok/s` prefill (`pp512`) and `16.10 tok/s` decode (`tg32`) with `-ngl 99 -fa 1 -b 512 -ub 128` (`8.9x` CPU decode `1.81 tok/s`, `+35.1%` over `-fa 0` `11.92 tok/s`), and `+12.5%` decode speedup for `k = 6` (`11.7 t/s`) vs. `k = 8` (`10.4 t/s`). Renku's own Arc 140V Vulkan throughput remains `UNMEASURED`.

---

4. Coding Loss Localization Diagnostics (`MEASURED`)

4.1 Paired `k = 8` vs. `k = 6` Coding Audit (26 Benign Cases, Speculation OFF)

To test whether restoring `top_k = 8` at inference recovers standalone coding lost under `k = 6`, we ran a paired 26-case coding evaluation (`20` `coding_unit_test` from `eval_code_01..20` + `6` `repo_fixture_bugfix` from `repo_fix_01..06`, `-n 3072`, speculation OFF). Loader stderr confirmed `--override-kv qwen35moe.expert_used_count=int:8` changed `n_expert_used: 6 → 8`.

Metric / Error Category (26 Benign Cases)	`k = 6` Control (`release_eval_preds.jsonl`)	`k = 8` Override (`k8_coding_preds.jsonl`)	Paired Delta (`k=8` vs. `k=6`)
`coding_unit_test` (`eval_code_01..20`)	2/20 (`17, 19`)	3/20 (`02, 14, 17`)	+1 (`+2` gains `02, 14`; `-1` loss `19`)
`repo_fixture_bugfix` (`repo_fix_01..06`)	**5/6** (only `01` failed)	2/6 (`04, 06` only)	**-3** (losses on `02, 03, 05`)
Combined 26-Case Pass Rate (Original Verifier)	**7/26 (26.9%)**	**5/26 (19.2%)**	**Net -2 -> KEEP `k = 6` CONTROL**
Combined 26-Case Pass Rate (Fixed Extractor)	**7/26 (26.9%)**	**5/26 (19.2%)**	**Exact 0 extractor recovery delta**
`MODEL_LOGIC_WRONG_ANSWER`	7 (`03, 04, 07, 08, 12, 16, 18`)	4 (`01, 04, 18, repo_fix_01`)	Edge-case assertion failures
`MODEL_RUNTIME_OR_API_ERROR`	4 (`02, 09, 10, 14`)	7 (`03, 09, 10, 12, 16, 19, repo_fix_03`)	e.g., `OrderedDict.first_item()`, `UnboundLocalError`, `KeyError`
`BUDGET_TRUNCATION_AT_N3072`	5 (`01*, 06, 11, 15, 20*`)	7 (`05, 06, 11, 13, 15, repo_fix_02*, 05*`)	`06, 11, 15` burned `9.9k–12.3k` chars in `<think>` w/ 0 answer
`MODEL_SYNTAX_ERROR`	2 (`13, repo_fix_01`)	3 (`07, 08, 20`)	Unterminated string literals, unclosed `[`
`MODEL_REPETITION_LOOP`	1 (`05`)	0	Degenerate multi-block repeat
Wall Time (26 cases)	`597.73 s`	`540.98 s`	Identical settings, speculation OFF
Mean Decode Throughput (`llama-cli`)	**`125.17 tok/s`**	`120.55 tok/s` (`0.9632x`)	**-3.68%** decode throughput
Verified Fixes / Hour	**`42.16 fixes/hr`**	`33.27 fixes/hr`	**-8.89 fixes/hr** (`pgmajfault_delta = 6`, `min_host_mem = 165,921 MiB`)

Verdict: NO-GO on `k = 8` promotion; KEEP `k = 6` CONTROL. Because all `40` `mlp.gate` routers were healed under `top_k = 6` softmax normalization during Phase 3A SFT, forcing `k = 8` at inference pulls in two uncalibrated lower-ranked experts, degrading `repo_fix` (`5/6 → 2/6`) and cutting verified fixes/hour by `21.1%` (`42.16 → 33.27`). A conditional 5-item vision check did not trigger because `k = 8` regressed on coding.

4.2 Zero-Artifact Forensics on `llama-cli` Coding Predictions

Re-scoring all `26` `k = 6` predictions and all `26` `k = 8` predictions with our hardened `verifiers.py` (supporting unmatched closing `</think>` tags and `entry_point`-aware AST-valid code block extraction) yielded **exact `0` recovery delta** on both arms (`7/26 → 7/26` and `5/26 → 5/26`).
`colab_parallel_release_eval.py::_clean_cli()` had already mapped `llama-cli`'s `[Start thinking]` / `[End thinking]` delimiters into paired `<think>` … `</think>` blocks. Consequently, all `19` `k = 6` failures represent genuine model errors—primarily logic bugs (`7`), hallucinated API/runtime errors (`4`), and `3,072`-token thinking exhaustion (`5`).

4.3 Six-Stage One-Variable-at-a-Time Localization Ladder and Sequence-Cap Effects

To isolate exactly where standalone coding capability (`2/20`) was lost, we executed a one-variable-at-a-time localization ladder (`eval/localization_ladder_manifest.json`) with speculation OFF and no deeper pruning or GRPO permitted until localization completed.

Stage	Artifact / Effective `k`	Single Variable Changed	`1,536`-Token Probe (Cell 15)	`3,072`-Token Confirmation (Cell 19) & Prior Measurements
1	Stock `Qwen/Qwen3.6-35B-A3B` BF16 (`256`, `k=8`)	Unpruned stock baseline	`BLOCKED_SIZE` (`71.927 GB` / `66.987 GiB` vs. `18.954 GiB` overlay free)	Matched Stock `Q4_K_M` (`20.543 GB` in `/dev/shm`): **`4/6`** on `repo_fix` (Cell 14)
2	`checkpoints/pruned_168` (`168`, `k=8`)	REAP `256 → 168` (BF16, raw unhealed)	**`0/6`** (`668.4 s`, `9.51 t/s`, `45,152 MiB`)	**`0/26` COMPLETE** (`0/20` `eval_code` + `0/6` `repo_fix`; severe `answer_loop` up to `1,016x` repeats + `3–7`-tok premature EOS)
3	`checkpoints/pruned_168` (`168`, `k=6`)	Routing `k=8 → k=6` (BF16, raw unhealed)	**`0/6`** (`808.4 s`, `10.00 t/s`, `45,142 MiB`)	**`0/10` observed live** (`9.94 t/s`, `6/26` in `09:19` Drive snapshot; remaining `16` items unrecovered after Colab idle recycle)
4	`checkpoints/renku_v1_sft_inherited_mtp` (`168`, `k=6`)	`+SFT LoRA + 40 unfrozen mlp.gate` (BF16)	**`0/6`** (0 loops/EOS — pure `1,536`-tok truncation)	**`4/26` COMPLETE** (`2/20` `eval_code` [`03, 17`], matching Q4's `2/20` + `2/6` `repo_fix` [`04, 06`]; **`9/20` `eval_code` still hit `3,072` cap**)
5	`exports/Kasui-Renku-1-F16.gguf` (`168`, `k=6`)	HF BF16 -> GGUF F16 (`48.338 GB`)	—	**`1/6`** on 6 localization misses (`0/3` coding — matches Q4 exactly; **zero quantization loss**)
6	`exports/Kasui-Renku-1-APEX-Q4-sub15G.gguf` (`168`, `k=6` vs. `k=8`)	F16 -> APEX Q4 (`14.879 GB`) + `mmproj-Q8_0`	—	**`k = 6`: `30/56`** (`7/26` coding = `2/20` + `5/6` `repo_fix`, `3/5` Gate-0 vision); **`k = 8`: `5/26`** (`3/20` + `2/6`)

Three diagnostic conclusions follow directly from Table 6:

1. **Zero quantization loss:** Comparing Stage 4 BF16 (`2/20` `eval_code`), Stage 5 F16 (`1/6` misses, `0/3` coding), and Stage 6 APEX Q4 (`2/20` `eval_code`, `1/6` identical misses) proves that APEX mixed-precision Q4 quantization is NOT responsible for the standalone coding deficit.
2. **Severe sequence-cap sensitivity:** At `1,536` tokens, even the healed BF16 checkpoint (Stage 4) scored `0/6` despite having zero loops and zero premature EOS—solely because `<think>` traces exhausted `1,536` tokens before emitting code. Raising the cap to `3,072` recovered `4/26`, yet `9/20` BF16 `eval_code` items (and `5/26` Q4 `k = 6` items) still exhausted `3,072` tokens. Auditing `v1` `train.jsonl` revealed **`72.7%`** in-think fenced-code bloat, directly explaining why the SFT model drafts full code implementations inside `<think>` before rewriting them in the final answer.
3. **Upstream BF16 pruning collapse + SFT contamination:** Raw unhealed `pruned_168` suffered total coding collapse (`0/26` at `k = 8`, `0/10` observed at `k = 6`) with up to `1,016x` repetition loops and `3–7`-token premature EOS. Phase 3A SFT + router healing fully cured looping/EOS and achieved `5/6` on `repo_fix` (Q4), but full-sequence prompt loss contamination (flat `"text"` rendering instead of assistant-only loss) and verbose in-think code duplication left algorithmic `eval_code` tasks at `2/20`.

---

5. Offline Remediation, Curated Candidates, and Fresh Held-Out Suite (`MEASURED` Offline)

Before authorizing any new GPU training, all diagnosed pipeline defects were repaired and verified strictly offline (`25/25` to `26/26` passing unit tests; zero GPU training launched; **`quality_gain_claim = UNCLAIMED_UNTIL_HELDOUT_AB`**):

- **Harness & verifier fixes (`verifiers.py`, `colab_parallel_release_eval.py`):** Added unmatched closing `</think>` detection for HF/OpenAI endpoints (where Qwen3.6's `chat_template.jinja` places `<think>\n` inside

the prompt) and `entry_point`-aware AST-valid Python block selection so trailing `def demo()` examples cannot shadow target functions.

- **Training & pruning fixes (`train_sft_grpo_mtp.py`, `prune_and_heal.py`):** Added `prepare_sft_records_and_loss_mode` (conversational `"messages"` + `assistant_only_loss = True` via Qwen3.6 `{%- generation %}` spans), unique `fixture_id` indexing fixing the `300 → 140` GRPO collision, and `compute_topk_router_energy` + `--calib-completions-file` so future REAP saliency masks to active top-k experts and calibrates on verified assistant code completions (`data/calib_v2_code_completions.jsonl`, `240` held-out-disjoint rows including `112` verified code completions).
- **Offline condensed and CPU-verified training candidates (strictly separate from v1, zero overwrites):**
  - *Condensed v1 (`train_v15_condensed_coding_candidate.jsonl`, `1,001` rows, `702,343` exact Qwen3.6 BPE tokens, `-56.62%` vs. `clean_4096`):* `83/83` independently unit-test-verified `coding_unit_test` + `300` SWE/multi-lang + `618` non-coding replay.
  - *Clean-Code Condensed v2 (`train_v15_condensed_clean_code_v2.jsonl`, `1,108` rows, `716,373` tokens):* `150/150` verified `coding_unit_test`, deterministically stripping `306` redundant in-think code fences (`0` remaining in coding) and `247` post-code essays.
  - *Low-Priority Colab CPU-Curated Pack (`train_v15_cpu_verified_coding_pack.jsonl`, `1,548` rows, `1,222,777` tokens, `wall = 124.09 s` at `nice -n 19 ionice -c 3` with `≤2.68%` impact on concurrent Cell 19 GPU throughput):* Scanned `9,043` existing Qwen4 Max rows, executed `1,186` isolated `python -I` unit-test runs (`1,186/1,186` passed) yielding **`238`** unique verified `coding_unit_test` problems (`+858` total coding + `690` replay), stripping `1,344` in-think code fences and `1,128` post-code essays. A companion quarantine ledger (`data/quarantine_v15_cpu_audit.jsonl`, `4,591` rows) caught `19` held-out `TTLLRUCache` leakage rows, `84` malformed unclosed `<think>` rows, and `27` multi-language verifier failures.
- **Fresh Exact-Test Held-Out Suite (`eval/fresh_heldout_coding_fixtures_v1.jsonl`):** `20` novel coding tasks (`14` standalone + `6` realistic bugfixes), `20/20` reference solutions PASS and `20/20` buggy baselines REJECTED in isolated `python -I` sandboxes, `0` entry-point collisions, `0` exact/normalized overlaps, `max_5gram_jaccard = 0.1159` against all train/seed/`eval_50`/`repo_fix` sets.

---

6. Pending Benchmarks, Resume Handoff, and Comparability Guardrails

6.1 Verified Durable Archive Handoff (`MEASURED`)

Before the initial Colab Pro A100 runtime reached its idle-inactivity recycle (~`40 CU` remaining), **`~66.32 GB`** of critical artifacts were readback SHA-256 verified to Google Drive (`/content/drive/MyDrive/Kasui-Renku-1-Backup`, `errors = 0`):

- **Tier 1 (`151/151` files, `0.721 GB`):** All scripts, tokenizer, `v1`/`v1.5`/`v2` datasets, Inkling traces, `imatrix.dat`, `checkpoints/sft/` (adapter + 40 `mlp.gate`), `checkpoints/v15_pilot_sft/`, fresh held-out fixtures, `bf16_confirm_3072_preds.jsonl` (`09:19 PST` snapshot), and telemetry.
- **Tier 2 (`3` files, `16.389 GB`):** `Kasui-Renku-1-APEX-Q4-sub15G.gguf` (`14.879 GB`), `mmproj-Q8_0.gguf` (`0.611 GB`), and `mmproj-F16.gguf` (`0.899 GB`).

- **Tier 3 (`26/26` shards, `49.212 GB`):** Complete `checkpoints/renku_v1_sft_inherited_mtp` BF16 checkpoint.

6.2 Pending Benchmarks and Verification Gates (`PENDING`)

Table 7 summarizes all unmeasured and queued release benchmarks per RENKU_EVALUATION_PROTOCOL.md and RENKU_COMPUTE_TRANCHES.md (3 x 200-CU checkpointed subscriptions with mandatory `≥15–20 CU` handoff buffers).

Benchmark / Verification Gate	Required Hardware & Protocol	Current Status	Blocking Requirement / Notes
Remaining Gate 0 Smoke	Shipped Q4 + `mmproj-Q8_0` (`A100-40GB` or local)	`PENDING`	Text + bar-chart smoke passed; real JSON tool-call & standalone invoice OCR gate remaining
Target-Hardware 16 GiB Fit (`32K + Vision`)	**Actual 16 GiB Intel Arc 140V Vulkan (`8a97fa9`)**	`PENDING_TARGET_HARDWARE`	Must verify `≥3 GiB` free system RAM, zero OOM, zero sustained warm paging (never infer from A100 CUDA)
Occupied 64K Long-Context Suite	Separate Large-Context cell (`65,536` occupied tokens + output headroom)	`PENDING`	Must test multi-key retrieval, variable tracing, aggregation, and conflicting errata (not empty `-c 65536`)
Sealed Held-Out Vision Suite	Disjoint from `calib_manifest.json` (`mmproj-Q8_0`)	`PENDING`	GDP.pdf-style document synthesis + `≥100–149` screenshots (OCR, charts, tables, small text)
Primary Pilot: Terminal-Bench 4.0	Fixed ~12 non-security tasks, pinned `mini-swe-agent` , `A100-40GB` (`~6.8 CU/h` assumption)	`PENDING`	Must measure empirical `minutes/task` (incl. load/teardown/idle) and `verified tasks/hr` before any 66 x 3 AA projection
Second Wave: TB-Science & SWE-bench Verified	Non-sensitive public subsets, documented budgets	`PENDING`	Queued strictly after Primary Pilot and CU verification
Matched Peer & Stock Comparison	Common `llama.cpp @ 8a97fa9` , identical seeds/harness/quant tier	`PENDING`	Optional post-release comparison vs. Stock `Qwen3.6-35B-A3B` and local 24 GB panel (`Qwen3.8-27B` , `Inkling-2-Mini` , etc.)
Optional `v1.5` Clean-Code LoRA A/B	Bounded `A100-80GB` run into disjoint checkpoint	`PENDING`	Must pass held-out coding improvement **AND** mandatory held-out vision non-regression (0 image pairs in `v1.5`)

6.3 Strict Comparability Guardrails and Rights Boundary

- **No undisclosed-precision equivalence against Qwen card rows:** Although `Qwen/Qwen3.6-35B-A3B` ships BF16 weights, exact inference precision per benchmark row on the official model card is undisclosed; treat published card scores strictly as background reference, never as matched controls.
- **Terminal-Bench 2.0 vs. 4.0 non-comparability:** Qwen3.6-35B-A3B's published Terminal-Bench 2.0 score (`51.5`) is **not** comparable to Terminal-Bench 4.0 (different tasks, version, and harness) and must never be diffed or plotted against TB 4.0 scores.
- **No Artificial Analysis v5 composite claims:** Public subsets must never be aggregated into an Artificial Analysis Intelligence Index v5 composite score.
- **Licensing and teacher-trace rights boundary:** While `Qwen3.6-35B-A3B` open weights are Apache-2.0, OpenRouter's pinned Alibaba provider links Alibaba Model Studio Terms of Service, which restrict using API outputs to train or develop competing products or services without express authorization. Spend ledgers (`cost_ledger*.json`) document API spend only; explicit licensing clearance is required before any external or competing release trained on API-distilled traces.

Sources and Primary Artifacts

- Qwen3.6-35B-A3B Model Card & Benchmark Results
- Artificial Analysis Intelligence Benchmarking Methodology
- `ggml-org/llama.cpp` (pinned commit `8a97fa98ab1a454709bbf5ac979661feaeadec40`)
- `SWE-agent/mini-swe-agent`
- Internal Audit & Protocol Records: CODING_LOCALIZATION_AUDIT.md, RENKU_EVALUATION_PROTOCOL.md, RENKU_COMPUTE_TRANCHES.md, COLAB_HANDOFF_AND_RESUME.md, export_apex_gguf.sh, apex_tensor_types.txt, prune_and_heal.py, and train_sft_grpo_mtp.py.