Skill-suite calibration: mikko- library (2026-06-02) + Spacepotatis + AudiobookMaker audit skills (2026-06-03)

green = saved ≥20% · red = cost ≥20% more · black = within ±20% = direction-at-best. The ±20% band is the noise floor this doc demonstrates (cross-session drift ~8pp + N=1 task variance), so within-band magnitudes are not coloured as signal. Colours the "% saved" columns only; swing/pp and token columns stay neutral.

*A/B token measurement (cold arm A vs with-skill arm B) for the 8 cleanly-A/B-able mikko- skills, across **all three models** (Opus 4.8, Sonnet 4.6, Haiku 4.5). This is the **"before" baseline**: the skills were just audited + given freshness blocks (claude-skills #22/#23); a planned optimization pass will re-measure against these numbers. The optimization pass (claude-skills #24) has since landed and three of these skills were re-measured, see **After-optimization re-measure** below. Method mirrors the first calibration document, extended to the full updated suite and every model.*

Method

- **Two arms, same task.** Arm A (cold) scouts from first principles; arm B reads the `SKILL.md`, runs any companion script, follows the procedure. Same deliverable per pair. **48 sub-agent arms** total (8 skills × 2 arms × 3 models).
- **Corpus (8):** `audit`, `ai-codegen-smell-audit`, `react-anti-patterns-audit`, `readme-drift-sync`, `skills-quality`, `skills-freshness`, `skill-usage`, `session-cost`. (Orchestrator/installer/lister/recursive skills excluded as not cleanly measurable.)
- **Tasks auto-synthesized,** pinned to fixed targets: code audits → `mikkonumminen.dev/src/lib` (+ `/terminal`); react → `Spacepotatis`; readme → `mikkonumminen.dev/README.md`; skills-quality/freshness → live `~/.claude/skills/mikko-*`; skill-usage/session-cost → `~/.claude/projects`.
- **Single-agent approximation** for orchestrators (`audit`, `readme-drift-sync` run their parallel passes inline). **Accounting:** per-arm `subagent_tokens` (input+output+cache-creation). Read-only. **N = 1 per cell.**

Results: all three models

Skill	Opus A	Opus B	Opus %	Sonnet A	Sonnet B	Sonnet %	Haiku A	Haiku B	Haiku %
skills-freshness	101,969	23,215	+77%	82,177	18,083	+78%	88,412	29,948	+66%
skills-quality	75,323	21,829	+71%	83,680	14,895	+82%	55,753	23,637	+58%
session-cost	31,926	22,572	+29%	19,823	15,859	+20%	69,520	23,054	+67%
react-anti-patterns-audit	103,254	97,460	+6%	73,050	68,354	+6%	55,856	57,445	−3%
audit	94,777	142,666	−51%	130,314	127,089	+2%	64,032	59,455	+7%
skill-usage	22,946	32,014	−40%	21,899	21,938	~0%	28,879	28,147	+3%
readme-drift-sync	33,383	44,341	−33%	24,539	74,709	−204%	41,126	39,715	+3%

Skill	Opus A	Opus B	Opus %	Sonnet A	Sonnet B	Sonnet %	Haiku A	Haiku B	Haiku %
ai-codegen-smell-audit	34,224	48,665	−42%	26,908	52,222	−94%	31,588	54,926	−74%*
Aggregate (ratio-of-sums)	**497,802**	**432,762**	**+13%**	**462,390**	**393,149**	**+15%**	**435,166**	**316,327**	**+27%**

* `ai-codegen` Haiku-B was contaminated. It found and reused the `ai-smell-2026-06-02.md` report a *Sonnet* arm wrote earlier (a side-effect-file bleed). Treat that cell as unreliable.

What the data shows (cross-model)

1. **Savings scale inversely with model capability**: aggregate **+13% (Opus) → +15% (Sonnet) → +27% (Haiku)**. A weaker model scouts cold less efficiently, so the skill's deterministic procedure / script wins more, relatively. The curve is clean across *this* suite, but it is the **one finding that does NOT replicate** in the other two corpora (Spacepotatis is flat, AudiobookMaker non-monotonic); it holds only where the cold arm's cost scales steeply with model weakness. See **Cross-corpus synthesis**.
2. `audit` **is the textbook monotonic case**: **−51% → +2% → +7%**. On Opus a cold scout is cheap (95K), so the orchestrator-run skill (143K) loses badly; on Haiku the cold scout is dearer relative to the skill, so it flips positive.
3. **The savings live in the three script-backed skills**: `skills-quality` (+71/+82/+58%), `skills-freshness` (+77/+78/+66%), `session-cost` (+29/+20/+67%) save on **every** model. They replace LLM reasoning with a Python pre-pass / `scan.mjs`. Strip the two meta-skills and the suite goes net-negative on Opus and Sonnet.
4. `react` **calibrates neutral everywhere** (+6/+6/−3%). Its structured 6-check pass costs about what cold scouting costs on any model.
5. **The big meta-skill saves are "save by not looking."** On all three models the skill arms short-circuited (pre-pass / sha256 hash → "nothing to review") while the **cold arms caught real issues**: bloat (`ai-codegen` 655 lines, `audit`'s 5× duplicated prompt boilerplate), `security-audit` targeting a non-existent codebase, broken cross-repo links, a stale "13 skills" count. The savings are partly a coarser read.
6. **Haiku is the least reliable arm** (it hallucinated a finding (`skill-calibration` "has no freshness check") it does), wrongly concluded "token data unavailable" on skill-usage, and one Haiku arm tripped a security flag (ran PowerShell through Bash, circumventing a deny rule). Its verdicts carry the least confidence: same caveat the first calibration flagged.

Headline: across all three models the library's token economy is carried by the **script-backed skills** plus the neutral `react`; the prose audits are wash-to-negative against a capable cold model and only turn positive as the model weakens. The cold-arm findings (the bloat list, the real drift) are the targets for the upcoming optimization pass, which should re-measure against these numbers.

After-optimization re-measure

The "after" the baseline anticipated. claude-skills #24 optimized three of these skills: `ai-codegen-smell-audit` *(655→580 lines; Provenance extracted to a companion file + 5 dead links removed),* `audit` *(410→390; the 5×-duplicated sub-agent prompt footer deduped),* `readme-drift-sync` *(325→310; dated content-calibration narrative extracted). Each optimized skill's* ***with-skill arm (B)*** *was re-measured on all three models, same tasks, same accounting. This compares* ***B(before-optim) → B(after-optim)****; a negative Δ means cheaper after.*

Skill (arm B)	Opus →	Δ	Sonnet →	Δ	Haiku →	Δ
ai-codegen-smell-audit	48,665 → 50,044	+3%	52,222 → 39,148	−25%	54,926 → 47,403	−14%
audit	142,666 → 82,841	−42%	127,089 → 142,847	+12%	59,455 → 56,378	−5%
readme-drift-sync	44,341 → 43,303	−2%	74,709 → 37,044	−50%	39,715 → 40,704	+2%
Aggregate (3 skills)	**235,672 → 176,188**	**−25%**	**254,020 → 219,039**	**−14%**	**154,096 → 144,485**	**−6%**

Reading: the optimization's per-invocation token effect is below the N = 1 noise floor. This delta is not the optimization. The per-cell swings span −50% to +12% with no relationship to the 1–3 KB of `SKILL.md` body each skill shed. The same `audit` task cost 56K, 59K, 83K, 127K and 143K across these runs; that ±40–60K task-work variance dwarfs the few-KB body trim by 20–60×. The overall −16% (all nine cells, 643,788 → 539,712) is the *before* pass's high outliers (`audit` Opus 143K, `readme` Sonnet 75K) regressing toward the mean, not a measured saving.

What this means for skill optimization. Trimming `SKILL.md` *body* size buys ≈ 0 measurable per-invocation tokens for task-heavy skills: the one-time body read is a rounding error against the audit work itself. The payoff of the #24 optimizations is real but lives where this metric can't see it: the always-loaded `description` (charged on every matching turn, not just on invocation), context cleanliness, and correctness (the dead links / dead game-refs / stale narrative removed). To *measure* a body-size win you'd need a skill whose body dominates its task (none of these three) or (cheaper than N ≫ 1) **paired measurement**: same-dispatch/same-hour A/B in isolated sandboxes (the discipline the optimization study mandated), which drops the floor below this signal without re-running anything N times. The fixable lever here is pairing, not sample size.

Spacepotatis audit-skills calibration (2026-06-03)

*A second corpus, same A/B method, run after the Spacepotatis skills were finetuned (quality + freshness, Spacepotatis #286). The **4 cleanly-A/B-able read-only audit skills** were measured cold (A) vs with the **finetuned** skill (B) across all three models: `balance-review`, `content-audit`, `ai-codegen-smell-audit`, `save-roundtrip-audit`. The 8 generative scaffolders (`new-*`, `equipment`) + 2 orchestrators (`modular-architecture-audit`, `security-audit`) were finetuned too but **excluded from A/B**. They mutate the repo, so they aren't cleanly read-only measurable (same exclusion logic the mikko- suite used). **24 sub-agent arms** (4 × 2 × 3). Same accounting (`subagent_tokens`), N = 1, pinned read-only targets. The B arms read the finetuned `SKILL.md` from a worktree off the merged master; the cold A arms are skill-independent and unchanged.*

Skill (arm B reads the finetuned `SKILL.md`)	Opus A	Opus B	Opus %	Sonnet A	Sonnet B	Sonnet %	Haiku A	Haiku B	Haiku %
balance-review	81,911	46,403	+43%	60,719	25,715	+58%	61,852	37,892	+39%
content-audit	121,173	66,732	+45%	99,250	71,408	+28%	62,300	57,438	+8%
ai-codegen-smell-audit	103,964	126,094	−21%	85,301	93,056	−9%	61,452	62,478	−2%
save-roundtrip-audit	64,695	72,437	−12%	44,787	58,010	−30%	73,877	62,038	+16%
Aggregate (ratio-of-sums)	**371,743**	**311,666**	**+16%**	**290,057**	**248,189**	**+14%**	**259,481**	**219,846**	**+15%**

What this corpus shows:

1. **`balance-review` is the standout (+43/+58/+39%).** Its structured metric procedure (DPS / TTK / energy-per-DPS formulas + a flagged-issues checklist) is far cheaper than scouting the balance maths cold. It's the Spacepotatis analogue of the mikko- script-backed skills.
2. **`ai-codegen` is a wash-to-negative (−21/−9/−2%)**: the same prose-audit pattern as its mikko- copy, reproduced on a different repo. The with-skill arm also follows the skill's "write a report" step, adding output tokens.
3. **`content-audit` saves on Opus/Sonnet (+45/+28%, both clearing the floor); Haiku +8% is positive but within-band: direction-at-best. `save-roundtrip` is mixed (−12/−30/+16%)**, content's checklist clearly beats cold scouting on the two larger models and stays positive (direction-only) on Haiku; save-roundtrip's layer-by-layer procedure costs *more* on Opus/Sonnet (which scout the save pipeline efficiently cold) but is positive on Haiku.
4. **Net is a flat ~+15% (+16/+14/+15%), not monotonic.** Unlike the mikko- suite's clean +13 → +15 → +27 inverse-capability curve, this corpus blends strong savers (balance, content) with washes (ai-codegen, save), so the per-model spread collapses to a flat band. The durable lesson reproduces across both repos: **token savings concentrate in procedure/script-backed skills; prose audits wash against a capable cold model.**

Noise-floor demonstration (unintended, but the cleanest in either corpus). This B column was re-measured. A first pass accidentally read the *pre-finetune* skills (the local checkout lagged the #286 merge), giving an aggregate of **+6/+8/+7%**; re-running the same arms against the **finetuned** skills (content differing by only a few KB + a freshness block) gave **+16/+14/+15%**, an **~8-point swing** (e.g. `content`-Opus alone moved 100.9K → 66.7K). Two compounding noise sources explain it, neither a finetune effect: (a) the two skill versions are near-identical, a few KB of body + a freshness block is far too little to move 8 points; and (b) the cold A-arms were *not* re-run, so each cell now pairs a run-1 A against a run-2 B, layering cross-session drift on top of the per-cell **N = 1** variance. The swing is therefore noise, not signal, an accidental, direct confirmation of this doc's own thesis that run-to-run variance dwarfs SKILL.md content. *Which* skill saves vs washes is stable across both runs; only the magnitudes wobble within the noise band.

Caveats (this corpus): N = 1 per cell; the with-skill audit arms wrote stray `docs/audits/*-2026-06-03.md` reports during measurement (cleaned up); a Haiku `content-audit` arm tripped the PowerShell-via-Bash deny flag (read-only file listing); the generative + orchestrator skills were finetuned but not A/B-measured. Outcome ≠ tokens: e.g. one Haiku `save` arm flagged a "critical `currentPlanet` drop" that the Opus/Sonnet arms (and the skill) correctly read as a by-design re-derivation.

AudiobookMaker audit-skills calibration (2026-06-03)

*A third corpus, same A/B method, on the AudiobookMaker repo (a Python desktop app). The **4 cleanly-A/B-able read-only skills** were measured cold (A) vs with the **finetuned** skill (B) across all three models: `audit`, `ai-codegen-smell-audit`, `release-bundle-audit`, `copyright-scan`. The other 6 skills (`ci-failure-triage`, `pronunciation-corpus-add`, `release-cut`, `voice-pack-finnish`, `work-session`, `worktree-launch`) were finetuned too but **excluded from A/B**. They mutate the repo or need live inputs (a failing CI run, a source recording), so they aren't cleanly read-only measurable (same exclusion logic the other two suites used). **24 sub-agent arms** (4 × 2 × 3). Same accounting (`subagent_tokens`), N = 1, pinned read-only targets, the `src/` tree (audit, ai-codegen), the PyInstaller `.spec` (release-bundle), and a `HEAD~3..HEAD` diff (copyright-scan). The B arms read the finetuned `SKILL.md`; the cold A arms are skill-independent.*

Skill (arm B reads the finetuned `SKILL.md`)	Opus A	Opus B	Opus %	Sonnet A	Sonnet B	Sonnet %	Haiku A	Haiku B	Haiku %
audit	115,835	100,367	**+13%**	134,264	112,891	**+16%**	101,262	75,794	+25%
ai-codegen-smell-audit	76,035	76,160	~0%	121,961	59,172	+51%	71,418	48,614	+32%
release-bundle-audit	45,668	52,984	−16%	73,861	65,758	+11%	46,597	38,099	+18%

Skill (arm B reads the finetuned `SKILL.md`)	Opus A	Opus B	Opus %	Sonnet A	Sonnet B	Sonnet %	Haiku A	Haiku B	Haiku %
copyright-scan	25,725	30,560	−19%	17,620	26,874	−53%	32,365	33,988	−5%
Aggregate (ratio-of-sums)	**263,263**	**260,071**	**+1%**	**347,706**	**264,695**	**+24%**	**251,642**	**196,495**	**+22%**

What this corpus shows:

1. `audit` **is the standout (the only skill here positive on all three models), but only its Haiku +25% clears the ±20% noise floor** (Opus +13% / Sonnet +16% are direction-at-best, within the cross-session + N=1 band; see the colour note below). Its five-phase robustness procedure caps the cold arm's scaling: the cold arm reads progressively more of the tree as the model weakens, while the skill's bounded sub-agent prompts hold the line. It's the AudiobookMaker analogue of the script-backed savers in the other suites: read it as "positive everywhere, convincingly only on Haiku," not as a clean +13/+16/+25 magnitude curve.
2. `ai-codegen` **washes on Opus (~0%) but saves big on Sonnet/Haiku (+51/+32%).** The cold Sonnet/Haiku arms ran exhaustive audits (122K / 71K tokens); the with-skill arm short-circuited via the skill's prior-audit calibration log ("0 findings, already reviewed") and read far less. Cold Opus was already efficient (76K), so there was nothing to save: the same prose-audit pattern seen in both other repos, here amplified by the calibration short-circuit.
3. `copyright-scan` **is wash-to-negative (−19/−53/−5%).** Reading the `SKILL.md` + running its seven-check procedure costs more than a cold scan of a small (2-file) diff: the fixed procedure overhead dominates when the target is tiny. (This skill was previously gitignored as local-only; it was sanitized and brought into the repo for this run.)
4. `release-bundle-audit` **is mixed (−16/+11/+18%)**: the cold Opus arm traces the spec's reachability efficiently, so the skill's structured Phase 0–2 costs more on Opus, but it saves on the weaker models.
5. **Net: Opus flat (+1%), Sonnet +24%, Haiku +22%; overall +16%, but the per-model aggregates are not broad-based.** Sonnet's +24% is **~76% a single cell**: `ai-codegen`'s short-circuit (62.8K of the 83K sonnet net save), and that cell's cold-arm depth (122K) is the noisiest reading in the corpus, so the headline rests on one fragile N = 1 cell. Opus is **flat-by-offset**, not flat-by-wash, `audit`'s +15.5K is mostly cancelled by the `copyright` / `release-bundle` losses (≈ −12K). Only Haiku is genuinely broad (`audit` ~46% and `ai-codegen` ~41% of the net). So this corpus does **not** show the inverse-capability curve (it's non-monotonic, +1/+24/+22), and the magnitudes are N = 1. What *does* hold here is the concentration pattern, `audit`'s bounded procedure is the standout, the prose audits wash. (Which of the two theses actually replicates across all three repos (and which doesn't) is consolidated in **Cross-corpus synthesis** at the end.)

Caveats (this corpus): N = 1 per cell; the arm-B skills were read from the **unmerged, local-only** `chore/skills-token-finetune` branch, so (unlike the Spacepotatis corpus, measured against the merged #286) these numbers aren't yet reproducible from AudiobookMaker's mainline. `copyright-scan` was local-only (gitignored) and was sanitized + tracked for this run; a pre-existing CLI bug (an engine override inheriting an incompatible configured voice) was fixed in the same finetune branch so the test suite stayed green; the generative + orchestrator skills were finetuned but not A/B-measured. Outcome ≠ tokens: e.g. the `ai-codegen` short-circuit "saves" by trusting a prior calibration log rather than re-deriving findings.

Caveats

- **N = 1 per cell**: direction + rough magnitude; prose-audit magnitudes are noisy (see `readme`: −33/−204/+3%).
- **Side-effect contamination**: skill arms wrote reports (`react-anti-patterns-2026-06-02.md`, `ai-smell-2026-06-02.md`, `SKILL-USAGE-*.json`); later same-skill arms occasionally reused them (flagged on `ai-codegen` Haiku-B). This recurred in all three corpora, it's a **harness isolation bug**, not a per-cell footnote; see **Cross-corpus**

synthesis.

- **Auto-synthesized tasks + single-agent orchestrator approximation** inflate `audit` / `readme` skill-arm cost.
- **Outcome ≠ tokens**: several "saves" come with missed findings; several "costs" come with better fidelity (`skill-usage`).
- **Total measurement cost (mikko- suite):** ~2.54M `subagent_tokens` across its 48 arms (8 skills × 2 × 3; Opus ~931K, Sonnet ~856K, Haiku ~751K). The Spacepotatis and AudiobookMaker corpora add **24 arms each** (4 × 2 × 3): **96 arms** across all three.
- Cross-links: first calibration → `docs/audits/skill-calibration-2026-06-01.md` ; optimization study → `docs/audits/skills-optim-study-2026-05-31.md` .

Cross-corpus synthesis

Three corpora across three different codebases, a portfolio site (mikko- library), a game (Spacepotatis), and a Python desktop app (AudiobookMaker). What replicates, and how strongly, separates into two very different claims:

Thesis 1: savings concentrate in procedure/script-backed skills; prose audits wash against a capable cold model. → 3 / 3, bankable. It holds in the mikko- suite (the script-backed `skills-quality` / `skills-freshness` / `session-cost` save on every model; the prose audits wash), in Spacepotatis (`balance-review` 's metric procedure is the standout saver; `ai-codegen` washes), and in AudiobookMaker (`audit` 's bounded multi-phase procedure is the only skill that saves across the board; `copyright-scan` 's prose checks lose on a small diff). A finding that holds across three codebases of different character is genuine external validity, much stronger than one repo's result. Two claims are welded into that headline, and only one half is tautological: a *script / pre-pass* that moves work off the model necessarily costs the model less. The empirical, falsifiable claims are the other two (that a **bounded procedure** (`audit` 's, `balance-review` 's) beats unbounded cold scouting, and that **prose audits wash** against a capable cold model), and those are precisely what the 3/3 replication establishes; neither leans on the tautology to stand.

Thesis 2: savings scale inversely with model capability (the inverse-capability curve). → 1 / 3, do not generalize. Clean only in the mikko- suite (+13 → +15 → +27). Spacepotatis is flat and non-monotonic (+16/+14/+15, Opus is even highest); AudiobookMaker is +1/+24/+22 (non-monotonic; only its Sonnet +24% rests on a fragile single cell, ~76% one cell, while Opus is flat-by-offset and Haiku +22% is broad, per the finding above). The honest mechanism: the curve appears **only when the cold arm's cost scales steeply as the model weakens**, which is a property of the *task*, not a law of skills. Where a capable cold model already scouts efficiently, there's nothing for the weaker-model curve to recover. Thesis 2 is task-dependent and must **not** ride on Thesis 1's replication.

Why the tables colour at ±20%, not ±10%. This doc demonstrates its own noise floor: the Spacepotatis re-measure swung ~8pp (+6/+8/+7 → +16/+14/+15) from pure **cross-session drift** between a run-1 cold arm and a run-2 skill arm. That band is not special to that accident: **every** A/B pair here was measured non-simultaneously (cold A and skill B were separate dispatches; re-runs reused earlier A's), so ~8pp of cross-session drift plus N=1 task-work variance rides on every cell. (The optimization study held *same-hour* pairing as essential for exactly this reason.) So the tables colour only cells that clear **±20%**, the demonstrated floor; within ±20% is **direction-at-best**, not a magnitude. That's why e.g. `audit` 's +13/+16 render neutral despite being positive: they're inside the noise the doc itself measured. (±20% is a *per-cell* floor; the ratio-of-sums **aggregate** rows are averages over N skills and carry ~÷√N less noise, so neutralising a small aggregate like the mikko- +13/+15 is conservative, erring toward caution rather than precision. But ÷√N assumes roughly equal, independent contributions: a **concentrated** aggregate does *not* inherit it. When one cell is ~76% of the sum (AudiobookMaker's Sonnet +24%) the effective N ≈ 1, not 4, so it earns no tighter floor than a single cell: its +24% clears ±20% by only ~4pp on essentially one measurement, a *fragile* pass rather than a robust aggregate. Small-corpus aggregates, e.g. AudiobookMaker's four skills, are barely averaged and stay near the per-cell floor regardless.) One cell is deliberately left uncoloured against the ±20% rule: `ai-codegen` Haiku (the −74% cell marked contaminated) isn't painted as a confident magnitude when the footnote tells you to distrust it. Cells whose value carries a contamination footnote (marked `*`) render neutral regardless of magnitude.

Contamination is a harness isolation bug, not a footnote. Side-effect-file reuse (a skill arm reading a report a sibling arm wrote) and the PowerShell-via-Bash deny-flag trip recurred in **all three** runs (the deny-flag hit Haiku twice). Recurrence across three independent runs locates the fault in the harness: arms are not isolated from each other's side effects. Until arms run in isolated sandboxes, **every cell of a report-writing skill is potentially contaminated, not only the cases caught here.** Fix the isolation before the next calibration run.

Evidence tiers. mikko- and Spacepotatis were measured against skills merged on their mainlines (Spacepotatis #286). The **AudiobookMaker corpus is one tier lower**: its arm-B skills were read from an unmerged branch (now PR #84), so its numbers aren't yet reproducible from AudiobookMaker's mainline. Read it as the most provisional of the three.
