SKILLCOVER: SELECTING COMPLEMENTARY AGENT SKILLS UNDER NOISY EVALUATION

Anonymous authors
Paper under double-blind review

ABSTRACT

When an agent can retain only k skills, ranking them by individual performance can waste slots on redundant capabilities. SkillCover greedily selects skills from a fixed pool to maximize baseline-relative task coverage on a calibration matrix. This classical submodular objective measures the best performance available in the retained library. We analyze selection under task sampling and rollout noise, and evaluate coverage and agent success separately. On real SkillsBench packages, greedy improves held-out matrix coverage over singleton ranking on GLM-5 ($+0.083$ at $k = 4$); results on three other models are mixed. A deliberately redundant MCPAgentBench pool and synthetic controls show when singleton ranking wastes the budget on overlap. Deployment on 14 new tasks does not establish a success advantage. A fresh-calibration diagnostic gives positive mean differences on two models at four repetitions, but intervals include zero and one comparison reverses at two repetitions. Whether coverage gains improve deployment remains unresolved.

1 INTRODUCTION

An agent team may accumulate reusable skills for browsing, document editing, database queries and verification, while a budget for validation, maintenance or disclosure allows it to retain only k. Ranking skills by individual performance can retain several that cover the same tasks, leaving other tasks underserved.

Consider three skills and a budget of two. The first two solve the same 0.51 fraction of tasks; the third solves the remaining 0.49. Ranking by individual success, which we call singleton ranking, keeps the two duplicates and covers 0.51 of the tasks. Keeping one duplicate and the complementary skill covers every task. A skill's value depends on what is already in the library.

Library coverage is the value of choosing the best retained skill or an always-available baseline for each task. A deployed agent must also discover, choose and execute useful skills. We therefore evaluate library coverage and agent success separately.

SkillCover selects skills by their additional task coverage (Figure 1). It evaluates every candidate and the baseline on calibration tasks, then repeatedly adds the skill that most improves the best available score. The objective is classical monotone submodular maximization (Nemhauser et al., 1978; Wei et al., 2015). We study how noisy evaluation affects this selection rule and whether the retained libraries help deployed agents.

We make three contributions:

- We formulate fixed-pool skill retention as baseline-relative coverage selection, comparing greedy selection with singleton ranking on the same calibration matrix.
- We separate task-sampling error from rollout noise in the population coverage guarantee and state the additional assumptions needed for nearest-winner routing.
- We evaluate held-out matrix coverage and mounted-agent success using real skill packages, new-task deployment and fresh calibration, with redundant pools and synthetic studies to examine the modeled mechanisms.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

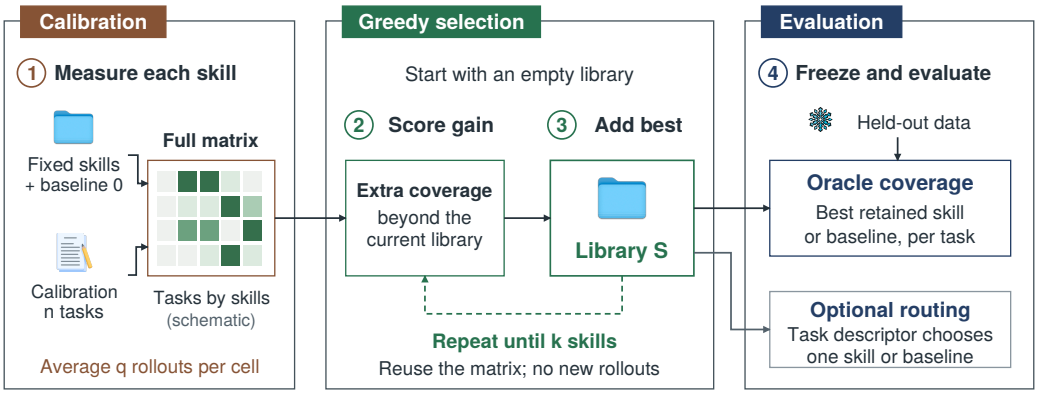


Figure 1: SkillCover's fixed-pool protocol. Estimate the full calibration matrix, then repeatedly add the skill with the largest coverage gain until k are retained. The dashed arrow returns the updated library to scoring; solid arrows carry data. Held-out coverage and optional routing are evaluated separately. Matrix shading is schematic.

Greedy improves coverage on GLM-5 and in the deliberately redundant pool, while the other Skills-Bench models give mixed results. Neither deployment study establishes a stable success advantage over singleton ranking.

2 SkillCover: Selecting Complementary Skills

The retention budget is an input, and the candidate pool is fixed before calibration. The objective evaluates at most one retained skill or the baseline per task. Each candidate is an atomic skill whose utility does not change when other skills are retained; composition and adaptive discovery require a different model.

Skills, tasks and utilities. Let $V = \{1, \dots, m\}$ be the candidate pool, let $0 \notin V$ be the always-available baseline, and let $1 \le k \le m$ be the retention budget. The baseline does not consume a retention slot. A task is $Z \sim P$. Let $Y_e(z) \in [0,1]$ be the score of a rollout using skill e, and let $u_e(z) = \mathbb{E}[Y_e(z)]$. Our real studies use normalized benchmark reward. The formulation also allows a reward–cost score transformed to $[0,1]$, provided the transformation is fixed before calibration (Appendix A).

Coverage objective. For a library $S \subseteq V$, its task-level oracle gain and population value are

$$f_z(S) = \max_{e \in S \cup \{0\}} u_e(z) - u_0(z), \qquad F(S) = \mathbb{E}_Z[f_Z(S)], \qquad |S| \le k. \tag{1}$$

For transformed reward–cost scores, clipping precedes baseline subtraction. Including the baseline makes $f_z(S) \in [0,1]$. The marginal gain of adding e is $\max(0, u_e(z) - \max_{a \in S \cup \{0\}} u_a(z))$. Thus a candidate contributes only where it improves on the current library. Its marginal gain decreases as that library improves, giving diminishing returns. Lemmas 1–2 prove boundedness and submodularity.

Full-matrix calibration. For each calibration task Z_i, skill $e \in V \cup \{0\}$, and repetition r, record $Y_{i,e,r} \in [0,1]$. The cell mean is $\hat{u}_{i,e} = q^{-1} \sum_{r=1}^{q} Y_{i,e,r}$. The empirical coverage is

$$\hat{F}_n(S) = \frac{1}{n} \sum_i \Big[\max_{e \in S \cup \{0\}} \hat{u}_{i,e} - \hat{u}_{i,0} \Big].$$

Collecting the full matrix requires $n(m+1)q$ rollouts. Both greedy and singleton ranking use this matrix without additional rollouts.

Algorithm 1 SkillCover: greedy selection from the full calibration matrix

Require: Cell means $\hat{u}_{i,e}$, candidate pool V, budget k, fixed tie rule
1: $S_0 \leftarrow \emptyset$
2: **for** $t = 0, \ldots, k-1$ **do**
3: $\quad e_t \in \arg\max_{e \in V \setminus S_t} [\hat{F}_n(S_t \cup \{e\}) - \hat{F}_n(S_t)]$, resolving ties by the fixed rule
4: $\quad S_{t+1} \leftarrow S_t \cup \{e_t\}$
5: **end for**
6: **return** $\hat{S}_g = S_k$

Greedy selection. Algorithm 1 adds the skill with the largest empirical marginal gain at each step. The reference implementation recomputes selected-column maxima; a current-maximum cache gives an $O(nmk)$ implementation. Appendix J distinguishes the two operation counts and records the implemented tie rules.

Singleton ranking. Our main comparator ranks candidates by individual coverage $\hat{F}_n(\{e\})$ and retains the top k, without recomputing their value after each selection. At the population level, it maximizes $F_{\text{singleton}}(S) = \sum_{e \in S} F(\{e\})$. This sum counts overlapping gains repeatedly, whereas greedy evaluates each candidate against the skills already retained.

3 Coverage guarantees and routing conditions

Complementarity under exact utilities. Including the baseline makes F normalized, monotone and submodular. Exact greedy therefore achieves the classical $1-1/e$ approximation to the best size-k library. Singleton ranking has no constant guarantee independent of k: a pool of duplicates and complementary skills can drive its ratio to the optimum arbitrarily close to $1/k$. These statements concern coverage under fixed, non-interacting skill utilities (Theorem 1).

Two sources of calibration error. For i.i.d. calibration tasks and conditionally unbiased cell estimates with $\sigma/\sqrt{q}$-sub-Gaussian error, Theorem 2 transfers the empirical greedy guarantee to population coverage. Its additive error scales as

$$O\left(\sqrt{\frac{k\log(em/k) + \log(1/\delta)}{n}} + \sigma\sqrt{\frac{\log((m+1)n/\delta)}{q}}\right)$$

with probability at least $1-\delta$. The first term controls task-sampling error uniformly over the candidate libraries; the second controls estimation error across all calibration cells. Increasing n reduces the first term, while increasing q reduces the second. Appendix A gives the exact bound and assumptions. The bound can be numerically loose (Appendix F.2) and does not determine which selector wins on a particular noisy matrix.

From coverage to execution. Nearest-winner routing chooses the skill with the highest estimated utility at the nearest calibration descriptor. Theorem 3 requires smooth descriptor-conditional utilities and estimates accurate for those conditional means. These may differ from the means for individual calibration tasks, so more repetitions alone do not ensure the routing condition. The reference is the best skill conditional on the descriptor; relating that value to task-level coverage also requires a bound on the information lost by the descriptor (Corollary 5). Our mounted-agent experiments use ordinary skill discovery, so their success is evaluated directly.

4 Evaluation: from library coverage to agent success

We first compare greedy with singleton ranking on real skill packages, then evaluate their retained libraries in mounted execution. A fresh-calibration diagnostic examines the sensitivity of that deployment comparison. Redundant pools and synthetic worlds then isolate overlap, evaluation noise and descriptor quality.

Study designs. The real-matrix studies split repeated rollouts on the same tasks, so they measure selection performance on separate rollout scores without testing new task identities. The transfer study evaluates frozen libraries on 14 new tasks. The fresh-calibration diagnostic uses new scores on disjoint calibration and evaluation task sets, but all task identities were previously seen. Synthetic studies draw independent tasks with known utilities.

Coverage metric. For real data, let $\bar{Y}_{i,e}^{\mathrm{hold}}$ be the held-out mean reward. We evaluate $\hat{F}_{\mathrm{hold}}(S) = n^{-1}\sum_i[\max_{e\in S\cup\{0\}}\bar{Y}_{i,e}^{\mathrm{hold}} - \bar{Y}_{i,0}^{\mathrm{hold}}]$. The coverage gap is the reference library's value minus the selected library's value; advantage is greedy minus the comparator selector. Exhaustive optimization gives the best library on the observed held-out matrix. That reference need not match the optimum of the unknown population objective: a maximum over one binary observation per skill can exceed the maximum of the skills' expected rewards. Synthetic oracle metrics use known utilities. We distinguish these coverage gaps from routing regret, which measures loss within a fixed library (worked examples: Appendix F).

4.1 Held-out coverage on real skill packages

We execute SkillsBench (Li et al., 2026) using OpenHands (Wang et al., 2024), with 30 tasks, a measured no-skill condition, and two repetitions across four base models. An entry-point audit excludes a license-only directory from the original 16 candidate directories, leaving 15 valid `SKILL.md` packages and 3,840 observations for selection analysis. Every model uses all 30 tasks, with no imputed rewards. Each candidate is evaluated individually to measure the coverage available in a retained set; Appendix G documents the complete archive and recovery of missing observations.

The comparators are singleton ranking, random subsets, farthest-first diversity on calibration mean and skill index, and greedy medoid selection over calibration columns. The primary comparison retains different historical tie rules: greedy keeps the earliest candidate in its scan order, while singleton uses reversed NumPy `argsort` (Appendix J).

We evaluate both rep0-to-rep1 and rep1-to-rep0 partitions and compare each selected library with exhaustive optimization on its held-out binary matrix. Task-bootstrap 90% intervals use split-averaged task advantages and condition on the fitted libraries. They omit re-selection uncertainty and correction for multiple budget comparisons. The statistic b_+ is the fraction of 2,000 bootstrap means strictly above zero, not a p-value or a posterior probability.

Coverage gains depend on the model and pool. GLM-5 has positive 90% advantage intervals at all tested budgets $k = 2, \ldots, 6$, and greedy matches the observed-matrix optimum at $k = 2, 3$. Qwen3-Coder and Sonnet do not show a resolved advantage at the two budgets in Table 1. Sonnet's point estimates are $+0.017$ at $k = 2$ and -0.033 at $k = 4$. DeepSeek-V3.2 has a negative $k = 4$ advantage, with interval $[-0.100, -0.017]$.

A control with common candidate priority on ties changes some comparisons. An exploratory leave-one-task-out refit also reverses GLM-5's positive same-task comparison (Appendix G). The evidence therefore supports coverage gains for some model–pool combinations under the stated evaluation protocol; the approximation factor does not predict the sign of every selector comparison.

Figure 2 shows the full budget sweep; domain summaries appear in Appendix G. The recorded per-model libraries differ, but different selected sets alone do not establish that a shared library performs worse.

4.2 Mounted execution on new tasks

Held-out matrix coverage chooses the best retained skill after seeing its score. Here we measure agent success with the library mounted before execution and its skills available for discovery and use.

Protocol. We fit four-candidate libraries using both repetitions of the 30-task calibration matrix, with common candidate priority for both selectors. The libraries are frozen before evaluation on 14 new tasks, two per original category. The four arms are Base, all 15 candidates, Greedy 4 and Singleton 4. All share an OpenHands background catalog; Base omits only our candidate pool.

Table 1: SkillsBench held-out matrix coverage relative to no-skill, using 15 valid skills and all 30 tasks per model (mean of two rollout partitions). Bold marks the best selector within each model and budget; OPT is the observed-matrix reference. Adv. = greedy minus singleton; b_+ is the bootstrap positive fraction.

Model	Greedy	Singleton	Diversity	k-med.	Random	OPT (ref.)	Adv.	90% CI / b_+
$k = 2$								
Claude Sonnet 5	0.150	0.133	**0.183**	0.150	0.142	0.217	+0.017	[+0.000, +0.050] / 0.63
GLM-5	**0.300**	0.183	0.233	0.183	0.180	0.300	+0.117	[+0.050, +0.200] / 1.00
Qwen3-Coder	**0.100**	**0.100**	0.050	0.050	0.098	0.183	+0.000	[−0.033, +0.033] / 0.36
DeepSeek-V3.2	0.100	0.100	**0.133**	**0.133**	0.108	0.250	+0.000	[−0.033, +0.033] / 0.34
$k = 4$								
Claude Sonnet 5	0.217	**0.250**	0.217	0.200	0.199	0.283	−0.033	[−0.117, +0.067] / 0.22
GLM-5	**0.350**	0.267	0.283	0.233	0.253	0.367	+0.083	[+0.017, +0.150] / 0.99
Qwen3-Coder	0.150	0.117	0.117	0.133	**0.153**	0.267	+0.033	[−0.050, +0.117] / 0.71
DeepSeek-V3.2	0.133	0.183	0.150	0.150	**0.184**	0.333	−0.050	[−0.100, −0.017] / 0.00

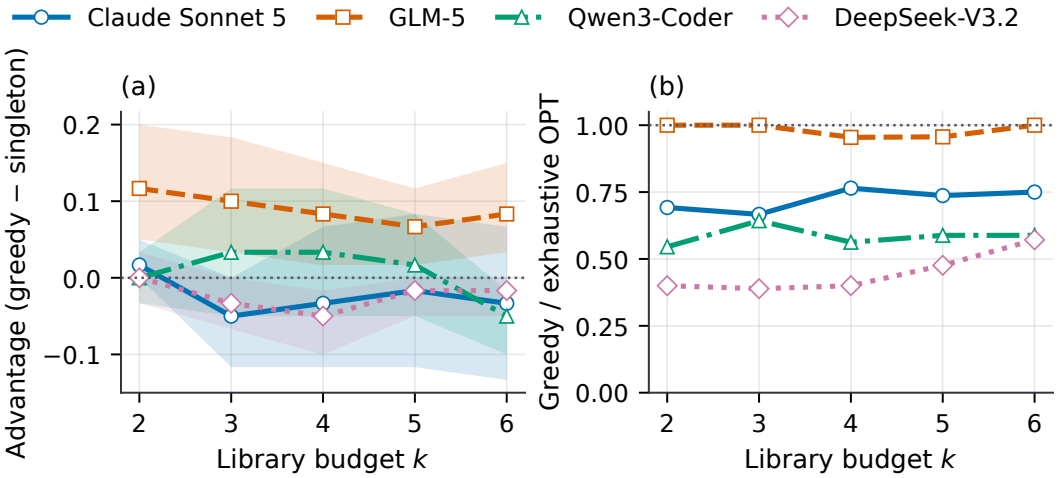


Figure 2: SkillsBench budget sweep, using all 30 tasks per model. (a) Greedy minus singleton advantage with conditional task-bootstrap 90% intervals. (b) Greedy value divided by the optimum of the observed held-out matrix. GLM-5 has positive advantage intervals throughout this sweep; the other models show mixed or unresolved outcomes.

Two repetitions require 420 unique rollouts because GLM's selectors choose the same set and share observations. Every planned execution receives a verifier score.

The success comparison remains unresolved. Table 2 reports full-credit success; partial credit is retained separately. Sonnet has a positive point difference of 0.107, Qwen 0.036, and DeepSeek −0.071, but all three paired 95% intervals include zero. These results do not establish superiority or equivalence. Appendix G.4 gives task sampling, native-score sensitivity, operational amendments and a post-hoc audit of skill exposure and use.

A historical mounted-library comparison on the original 30 tasks also leaves selected-versus-full success unresolved. Its configurations include a license-only directory and a three-skill selected arm for Qwen, so it is reported separately in Table 8.

4.3 Fresh Calibration and Deployment Variability

Protocol. We collect 768 fresh rollouts: two models, six calibration tasks, 15 individual candidates plus Base, and four repetitions. Both selectors use the same matrix and tie priority at $k = 4$. We fit libraries from either the first two repetitions ($q = 2$) or all four ($q = 4$), then evaluate on six separate document tasks with two repetitions. All task identities were previously seen, making this

Table 2: Mounted-library full-credit successes on 14 new tasks, with two repetitions per arm. Base omits our candidate pool; all arms retain 64 OpenHands background skill entries. Differences compare Greedy with Singleton; 95% intervals resample whole tasks and are unadjusted. † GLM's identical libraries share observations.

Model	Base	All 15	Greedy 4	Singleton 4	$\Delta_{\mathrm{G-S}}$	95% CI
Claude Sonnet 5	14/28	14/28	13/28	10/28	$+0.107$	$[-0.036, +0.250]$
GLM-5	4/28	8/28	4/28	4/28	$0^{\dagger}$	shared†
Qwen3-Coder	1/28	1/28	2/28	1/28	$+0.036$	$[-0.071, +0.143]$
DeepSeek-V3.2	1/28	2/28	2/28	4/28	-0.071	$[-0.214, +0.000]$

Table 3: Fresh-calibration diagnostic: success counts on six evaluation tasks, two repetitions each. Native rewards are binary; differences are mean Greedy-minus-Singleton rewards. $q = 4$ is primary. Intervals resample whole tasks, conditional on fitted libraries. Base is shared across q; † Sonnet's identical greedy libraries also share observations.

Model	q	Base	Greedy	Singleton	$\Delta_{\mathrm{G-S}}$	95% interval
DeepSeek-V3.2	2	1/12	0/12	1/12	-0.083	[-0.250, 0.000]
DeepSeek-V3.2	4	1/12	1/12	0/12	+0.083	[0.000, +0.250]
Sonnet	2	3/12	4/12†	3/12	+0.083	[-0.167, +0.333]
Sonnet	4	3/12	4/12†	2/12	+0.167	[0.000, +0.333]

an exploratory diagnostic. Sonnet's greedy libraries coincide across q, giving 108 unique evaluation executions.

Primary differences and repetition sensitivity. Table 3 shows positive primary $q = 4$ differences for DeepSeek ($+0.083$) and Sonnet ($+0.167$), with descriptive 95% intervals including zero. At $q = 2$, DeepSeek's difference is -0.083 and Sonnet's is $+0.083$. DeepSeek's $q = 4$ greedy success equals Base, and Sonnet exceeds Base by one success.

Both selectors tie in fitted coverage at $q = 4$, so the deployment differences do not demonstrate a fitted coverage gain. Comparing $q = 2$ with $q = 4$ also changes library identities and cannot isolate noise reduction. After image-size API failures, all eight Sonnet OCR executions were repeated with uniform image scaling; original scores remain separate. Appendix G.5 gives both configurations, task outcomes and uncertainty limits.

4.4 Coverage under constructed redundancy

To isolate how overlap affects selection, we use released MCPAgentBench score files (MCPAgentBench contributors, 2025): 31 model–setting pairs, 180 tasks and four runs, with correctness normalized from $\{0, 1, 2, 3\}$ to $[0, 1]$. Here a candidate is a model–setting pair. In each split, the weakest pair on calibration runs is the baseline and the remaining $m = 30$ pairs are candidates. Pooling settings with 10, 20, 30 and 40 distractor tools creates near-clones by design. We compare this redundant pool with a single-setting control, using only released scores.

Protocol. We enumerate all six ways to assign two runs to calibration and two to evaluation on the same tasks. The baseline and selected libraries use calibration data only. We report mean $\pm$ SE across splits as a descriptive summary: the splits share runs and are not independent replicates. The held-out reference is exhaustive for $k \leq 5$ and greedy-on-holdout for larger budgets (†).

Redundancy and budget dependence. At $k = 2$, greedy's held-out coverage gap is 0.006 ± 0.002, compared with 0.024 ± 0.005 for singleton ranking, a $4.1\times$ ratio and a positive advantage in five of six splits. Greedy attains 97.2% of the observed-matrix optimum. The advantage becomes small at larger budgets and changes sign at $k = 7, 8$, where the libraries largely overlap (Tables 16–17, Appendix H). The zero sampled submodularity-violation rate is an implementation check; submodularity follows from the objective's definition.

In the single-setting control ($m = 11$), the $k = 2$ advantage is -0.002 ± 0.004, positive in one of six splits. It becomes modestly positive at intermediate budgets and vanishes again by $k = 8$. The contrast supports redundancy as an explanation for this pool, without implying gains in every pool. The earlier baseline selected using all runs is retained only as a disclosed sensitivity analysis.

Appendix H also reports nearest-winner routing with coarse descriptors and retrospective calibration subsampling. These diagnostics address execution and acquisition separately from the coverage comparison.

4.5 Synthetic checks of redundancy, noise and routing

The clone construction in Theorem 1 yields greedy/OPT $= 1.0$, while the archived singleton/OPT result reaches 0.2052 at $k = 5$, near the limiting $1/k$. Independent-task latent-world sweeps over 30 seeds give mean greedy oracle coverage gap 0.0106, with advantages 0.1340 over singleton and 0.1208 over random selection (Table 20, Figures 4 and 3). These worlds illustrate the modeled mechanism without establishing its prevalence in deployed skill libraries.

Separate n- and q-sweeps have fitted log–log slopes approximately -0.40 and -0.50. One path injects Gaussian cell noise scaled by $1/\sqrt{q}$, so these are qualitative scaling checks rather than validation of literal rollout constants (Figure 5). In the fixed-library routing diagnostic, regret rises from 0.013 under clean descriptors to 0.178, 0.135, and 0.074 under random, incomplete, and aliased descriptors. Appendix I gives the diagnostics, and Appendix K records the limits of the archived evidence.

5 Related Work

Submodular maximization and coverage. The classical greedy analysis of Nemhauser et al. (1978) establishes an approximation guarantee for monotone submodular maximization under a cardinality constraint. Its $1 - 1/e$ factor is optimal unless P=NP (Feige, 1998). Related objectives appear in machine-learning subset selection (Das & Kempe, 2011; Wei et al., 2015). Work on uncertain evaluations includes robust selection (Krause et al., 2008), approximately submodular objectives (Horel & Singer, 2016), noisy-oracle greedy (Hassidim & Singer, 2017), noise-robust local search (Huang et al., 2022), and a unified treatment (Bhawalkar et al., 2025). SkillCover applies classical coverage selection to fixed agent-skill pools. Theorem 2 combines concentration and greedy transfer for task sampling and rollout noise in the calibration matrix; Theorem 1 illustrates the redundancy loss of singleton ranking.

Model routing, cascades, and ensembling. Model routing chooses which model to call for a query. Approaches include cost-aware API selection and cascades (Chen et al., 2020; 2024), learned routers (Ong et al., 2025; Ding et al., 2024), self-verification mixing (Aggarwal et al., 2024), and pairwise-ranking ensembles (Jiang et al., 2023). These systems optimize the per-query decision, typically without a budget on how many candidates remain available. SkillCover addresses the earlier decision: which k skills to retain from a frozen pool. Theorem 2 concerns the coverage available in that library; Theorem 3 bounds routing loss under additional assumptions. A learned router can use the selected library, but its deployed performance is a separate question.

Skill, tool, and prompt systems for LLM agents. Tool invocation during generation (Schick et al., 2023), lifelong skill acquisition and retrieval (Wang et al., 2023; Tziafas & Kasaei, 2024), and prompt search and selection (Zhou et al., 2023; Rubin et al., 2022) study how agents discover and adapt capabilities, often by combining skills. SkillCover selects a retained subset before per-task dispatch. With fixed atomic candidates, its objective in equation 1 is a baseline-relative facility-location function over task–skill utilities. ALFWorld (Shridhar et al., 2021) and MCP tool-use benchmarks (Wang et al., 2025; Guo et al., 2025) provide settings for evaluating agent capabilities. We directly run SkillsBench `SKILL.md` packages (Li et al., 2026) in Section 4.1 and use the score matrix released by the MCPAgentBench repository (MCPAgentBench contributors, 2025) for the redundancy study in Section 4.4.

Benchmark-subset selection and active evaluation. Active testing (Kossen et al., 2021) and benchmark compression (Polo et al., 2024; Vivek et al., 2024; Perlitz et al., 2024) select a few

informative examples to estimate model quality. Data subset selection for training uses coresets and related submodular criteria (Mirzasoleiman et al., 2020; Killamsetty et al., 2021). Our calibration-subsampling study (Appendix H.1) samples tasks within identifier groups and averages their scores to reduce the measurements used for library selection.

6 Discussion and Conclusion

SkillCover retains skills according to the tasks they add to the current library's coverage. It avoids the redundancy loss of singleton ranking in constructed pools and improves held-out matrix coverage on GLM-5 under the same-task rollout protocol. The mixed results on other models show that the benefit depends on the model, pool and evaluation design. The classical approximation factor compares greedy with the best coverage library; it guarantees neither a gain over singleton on every sampled matrix nor deployed agent success.

Held-out coverage tests whether selection preserves useful alternatives, while mounted execution tests whether those alternatives help the agent complete tasks. The 14-new-task study leaves the deployment comparison unresolved. Fresh calibration produces positive primary means on two models, but fitted coverage values tie, intervals include zero and the DeepSeek comparison reverses with fewer repetitions. These observations do not identify the cause of the deployment differences. Retained directory counts and hindsight single-skill maxima also do not measure token savings or identify routing failure.

Limits of the evidence. The real-matrix partitions reuse tasks, the transfer study has 14 category-balanced tasks, and the fresh-calibration diagnostic uses six previously seen evaluation tasks. Bootstrap intervals condition on fitted libraries. The diagnostic's OCR image policy was revised after outcomes were available, and both deployment studies retain background skills. The transfer selectors have no recorded explicit skill invocations; the fresh-calibration diagnostic records some, but the counters do not capture all possible use of skill content. These limits leave broad task generalization and the mechanism of skill use unresolved.

The fixed-pool model accounts for overlap when retaining skills. Extending it to interacting skills, composition or adaptive discovery changes the selection problem. Open questions are how to use the available coverage reliably and how to acquire enough evidence for selection without measuring every task–skill cell.

Reproducibility Statement

The source package provides selection code, archived results and SkillsBench observation bundles with their pre-recovery snapshots. Deterministic exports regenerate the real-skill tables and budget plot. Separate bundles retain all 420 transfer outcomes and the final diagnostic's 768 calibration and 108 evaluation outcomes, with task and library identities, protocol amendments, source hashes and all-attempt cost accounting. Table generation rechecks fitted libraries, means and bootstrap intervals from saved observations, without model calls. Appendices G.4 and G.5 document exposure audits and recovery procedures. MCPAgentBench uses third-party result files. Synthetic runners and outputs cover the main selection studies, but some legacy appendix diagnostics retain only plots and summaries; their full regeneration is not claimed. Appendix J lists code paths and available evidence. Compact bundles omit transcripts and credentials.

AI Use Statement

Generative AI tools assisted hypothesis refinement, experimental design and diagnostics, development of evaluation and analysis code, and interpretation of results. They also assisted manuscript structuring, prose drafting and revision, notation checks, reference verification, and examination of mathematical derivations. These uses included planning calibration and transfer comparisons and auditing saved rollout records. AI-assisted changes were checked through saved-observation reanalysis, deterministic table regeneration, regression checks for missing-data handling, bibliography source checks, and review of the compiled document. The experimental LLM agents are objects of

measurement: SkillsBench rewards come from deterministic benchmark verifiers, and MCPAgent-Bench scores come from its released result files. The authors take responsibility for the paper and for validating its AI-assisted content.

REFERENCES

Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, Shyam Upadhyay, Manaal Faruqui, and Mausam. AutoMix: Automatically mixing language models. In *Advances in Neural Information Processing Systems*, 2024.

Kshipra Bhawalkar, Yang Cai, Zhe Feng, Christopher Liaw, and Tao Lin. A unified approach to submodular maximization under noise. *arXiv preprint arXiv:2510.21128*, 2025.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalML: How to use ML prediction APIs more accurately and cheaply. In *Advances in Neural Information Processing Systems*, 2020.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024.

Abhimanyu Das and David Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. Hybrid LLM: Cost-efficient and quality-aware query routing. In *International Conference on Learning Representations*, 2024.

Uriel Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

Zikang Guo, Benfeng Xu, Chiwei Zhu, Wentao Hong, Xiaorui Wang, and Zhendong Mao. MCP-AgentBench: Evaluating real-world language agent performance with MCP-mediated tools. *arXiv preprint arXiv:2509.09734*, 2025.

Avinatan Hassidim and Yaron Singer. Submodular optimization under noise. In *Proceedings of the 2017 Conference on Learning Theory*, pp. 1069–1122, 2017.

Thibaut Horel and Yaron Singer. Maximization of approximately submodular functions. In *Advances in Neural Information Processing Systems*, 2016.

Lingxiao Huang, Yuyi Wang, Chunxue Yang, and Huanjian Zhou. Efficient submodular optimization under noise: Local search is robust. In *Advances in Neural Information Processing Systems*, 2022.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. LLM-Blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. GLISTER: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8110–8118, 2021.

Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Tom Rainforth. Active testing: Sample-efficient model evaluation. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Andreas Krause, H. Brendan McMahan, Carlos Guestrin, and Anupam Gupta. Robust submodular observation selection. *Journal of Machine Learning Research*, 9:2761–2801, 2008.

Xiangyi Li, Yimin Liu, Wenbo Chen, Bingran You, Zonglin Di, Yifeng He, Shenghan Zheng, et al. SkillsBench: Benchmarking how well agent skills work across diverse tasks. arXiv preprint arXiv:2602.12670, 2026. URL `https://arxiv.org/abs/2602.12670`.

MCPAgentBench contributors. MCPAgentBench: Evaluation results for tool-using LLM agents under distractor-tool settings. GitHub repository, https://github.com/Brunestuder/MCPAgentBench, 2025. Results snapshot at commit 96cfaa6, accessed August 2026.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14:265–294, 1978.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *International Conference on Learning Representations*, 2025.

Yotam Perlitz, Elron Bandel, Ariel Gera, Ofir Arviv, Liat Ein-Dor, Eyal Shnarch, Noam Slonim, Michal Shmueli-Scheuer, and Leshem Choshen. Efficient benchmarking (of language models). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, 2024.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinyBenchmarks: Evaluating LLMs with fewer examples. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics*, 2022.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021.

Georgios Tziafas and Hamidreza Kasaei. Lifelong robot library learning: Bootstrapping composable and generalizable skills for embodied control with language models. In *IEEE International Conference on Robotics and Automation*, pp. 515–522, 2024.

Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. Anchor points: Benchmarking models with much fewer examples. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, 2024.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang H. Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Yanjun Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, Junyang Lin, Robert Brennan, Hao Peng, Heng Ji, and Graham Neubig. OpenHands: An open platform for AI software developers as generalist agents. arXiv preprint arXiv:2407.16741, 2024.

Zhenting Wang, Qi Chang, Hemani Patel, Shashank Biju, Cheng-En Wu, Quan Liu, Aolin Ding, Alireza Rezazadeh, Ankit Shah, Yujia Bao, and Eugene Siow. MCP-Bench: Benchmarking tool-using LLM agents with complex real-world tasks via MCP servers. *arXiv preprint arXiv:2508.20453*, 2025.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1954–1963, 2015.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *International Conference on Learning Representations*, 2023.

A Formal coverage and routing guarantees

We state guarantees for exact coverage, noisy calibration and conditional routing under the assumptions below. Proofs begin in Appendix B; Section 4 evaluates coverage and execution separately.

A.1 Utility and routing definitions

A rollout score can combine reward and specified costs, $R_e(z) = r_e(z) - c_e^{\text{step}}(z) - c_e^{\text{token}}(z) - c_e^{\text{api}}(z) - c_e^{\text{tool}}(z) - c_e^{\text{invalid}}(z) - c_e^{\text{skill}}$. With $a > 0$ and b fixed before calibration, define $Y_e(z) = \text{clip}(aR_e(z) + b, 0, 1)$ and $u_e(z) = \mathbb{E}[Y_e(z)]$. Our real studies use normalized benchmark reward; they do not estimate every possible cost term.

The cell mean is $\hat{u}_{i,e} = q^{-1} \sum_{r=1}^{q} Y_{i,e,r}$. Define

$$F_n(S) = \frac{1}{n} \sum_i \big[\max_{e \in S \cup \{0\}} u_e(Z_i) - u_0(Z_i) \big], \qquad \hat{F}_n(S) = \frac{1}{n} \sum_i \big[\max_{e \in S \cup \{0\}} \hat{u}_{i,e} - \hat{u}_{i,0} \big].$$

We assume the cell estimator is conditionally unbiased with $\sigma/\sqrt{q}$-sub-Gaussian error. Independent bounded rollouts with the same conditional mean allow $\sigma = 1/2$. The full matrix fixes every empirical marginal; no extra rollout is requested during greedy selection.

Optional routing. A router observes a task descriptor X and chooses one retained skill or the baseline. Write $v_e(x) = \mathbb{E}[Y_e \mid X = x]$, $F_v(S) = \mathbb{E}[\max_{e \in S \cup \{0\}} v_e(X) - v_0(X)]$, and $J_v(\rho, S) = \mathbb{E}[v_{\rho(X)}(X) - v_0(X)]$. The selection guarantee concerns F; connecting it to deployed value J_v requires descriptor and routing assumptions.

A.2 Assumptions and theorem statements

Assumption 1 (Fixed atomic pool)**.** *The candidate pool $V = \{1, \dots, m\}$ is fixed before calibration. Each task invokes at most one retained skill or the baseline. Adding one skill does not change the utilities of any other skills.*

Assumption 2 (Calibration sampling)**.** *The calibration tasks are i.i.d., $Z_1, \dots, Z_n \overset{\text{iid}}{\sim} P$. For every calibration task Z_i, every skill $e \in V \cup \{0\}$, and every rollout $r \in \{1, \dots, q\}$, the rollout utility satisfies $Y_{i,e,r} \in [0, 1]$; the cell mean satisfies $\mathbb{E}[\hat{u}_{i,e} \mid Z_i] = u_e(Z_i)$, and the cell error $\hat{u}_{i,e} - u_e(Z_i)$ is conditionally $\sigma/\sqrt{q}$-sub-Gaussian.*

Assumption 3 (Descriptor-smooth deployment)**.** *After selection, each task has a descriptor $X \in \mathcal{X}$, where $(\mathcal{X}, d)$ is a metric space. For each retained skill $e \in S \cup \{0\}$, the descriptor-conditional utility $v_e(x) = \mathbb{E}[Y_e \mid X = x]$ is L-Lipschitz on the evaluated support: $|v_e(x) - v_e(x')| \le L\, d(x, x')$.*

Theorem 1 (Coverage, greedy approximation, and singleton redundancy)**.** *Suppose Assumption 1 holds, and define $F(S) = \mathbb{E}_Z[\max_{e \in S \cup \{0\}} u_e(Z) - u_0(Z)]$. Then F is normalized ($F(\emptyset) = 0$), monotone ($F(A) \le F(B)$ for $A \subseteq B \subseteq V$), and submodular: $F(A \cup \{e\}) - F(A) \ge F(B \cup \{e\}) - F(B)$ for $A \subseteq B \subseteq V$ and $e \in V \setminus B$. Let S_g be the exact greedy solution for the population objective F under $|S| \le k$. Then*

$$F(S_g) \ge \left(1 - \frac{1}{e}\right) \max_{|S| \le k} F(S).$$

Moreover, for every $k \ge 2$ and every $\eta > 0$, there exists a binary coverage instance with $u_0(z) = 0$ such that top-k singleton ranking attains at most $1/k + \eta$ of the optimal size-k value: in the clone construction, singleton ranking obtains $F(S_{\text{singleton}}) = p$ whereas the optimal library obtains $F(S^\star) = p + (k-1)q$, and the ratio $p/(p + (k-1)q)$ tends to $1/k$ as $p/q \to 1$.

The proof (Appendix B) averages the diminishing-returns inequality of Lemma 2 over tasks and applies the classical greedy theorem of Nemhauser et al. (1978). In the clone construction, p and q are task-region masses; this use of q is separate from the rollout count in the calibration protocol. Since $p > q$, the separation does not depend on ties.

Theorem 2 (Population coverage under noisy calibration)**.** *Suppose Assumptions 1 and 2 hold. Let* $N_k = \sum_{j=0}^{k} \binom{m}{j}$*, let* $a_g = 1 - 1/e$ *be the greedy approximation factor, and define the task-sampling and rollout-noise terms*

$$\beta_n = \sqrt{\frac{\log(4N_k/\delta)}{2n}}, \qquad \beta_q = \sigma\sqrt{\frac{2\log(4(m+1)n/\delta)}{q}}.$$

Let $\hat{S}_g$ *be the* k*-step greedy solution for* $\hat{F}_n$*. Then, with probability at least* $1-\delta$*,*

$$\sup_{|S|\le k} \left|\hat{F}_n(S) - F(S)\right| \le \beta_n + 2\beta_q,$$

$$\text{and, on the same event,} \qquad F(\hat{S}_g) \ge a_g \max_{|S|\le k} F(S) - (1+a_g)(\beta_n + 2\beta_q). \tag{2}$$

Equivalently, the excess term scales as $O\big(\sqrt{(k\log(em/k) + \log(1/\delta))/n} + \sigma\sqrt{\log((m+1)n/\delta)/q}\big)$.

The proof (Appendix C) bounds rollout noise by $2\beta_q$ (Lemma 5) and task-sampling error by β_n (Lemma 4). Because $\hat{F}_n$ remains normalized, monotone, and submodular, Lemma 6 transfers its greedy guarantee to the population objective.

This worst-case bound can be numerically loose (Appendix F.2). Section 4 therefore examines scaling trends and held-out gaps, rather than treating the bound as a precise performance prediction.

Theorem 3 (Nearest-winner routing under smooth descriptor-conditional utilities)**.** *Suppose Assumption 3 holds for a fixed selected library* S*. Let* $x_1, \dots, x_n$ *be the calibration descriptors, and for a test descriptor* x *let* $i(x) \in \arg\min_{1\le i\le n} d(x, x_i)$ *be the nearest calibration index. Assume the calibration winner estimates satisfy* $\max_{i,\, e\in S\cup\{0\}} |\hat{u}_{i,e} - v_e(x_i)| \le \epsilon_{\mathrm{desc}}$*, and define nearest-winner routing by* $\rho_S(x) \in \arg\max_{e\in S\cup\{0\}} \hat{u}_{i(x),e}$*. Then, for every test descriptor* x*,*

$$v_{\rho_S(x)}(x) \ge \max_{e\in S\cup\{0\}} v_e(x) - 2L\, d(x, x_{i(x)}) - 2\epsilon_{\mathrm{desc}}.$$

Consequently, with $F_v(S) = \mathbb{E}[\max_{e\in S\cup\{0\}} v_e(X) - v_0(X)]$ *the descriptor-conditional oracle value and* $J_v(\rho_S, S) = \mathbb{E}[v_{\rho_S(X)}(X) - v_0(X)]$ *the deployed routed utility,*

$$J_v(\rho_S, S) \ge F_v(S) - 2L\,\mathbb{E}\Big[\min_i d(X, x_i)\Big] - 2\epsilon_{\mathrm{desc}}.$$

The proof (Appendix D) compares utilities at a test descriptor and its nearest calibration descriptor. Applying the Lipschitz and calibration-error bounds in both directions gives loss at most $2L\, d(x, x_{i(x)}) + 2\epsilon_{\mathrm{desc}}$; averaging gives the deployed bound.

Without smooth, informative descriptors, Theorem 3 does not turn oracle coverage into a deployment guarantee.

Remark 1. *Theorem 2 controls task-level coverage* $F(\hat{S}_g)$*. Theorem 3 compares routed utility* $J_v(\rho_S, S)$ *with descriptor-conditional coverage* $F_v(S)$*. A descriptor may lose task information, so combining the results requires a bound on* $F(S) - F_v(S)$*, as in Corollary 5.*

Scope and limits. The selection results assume a fixed pool, a full calibration matrix and utilities that do not change when other skills are retained. The routing result additionally assumes a fixed library and suitable descriptors. Section 4 separates synthetic checks of these assumptions from the real-matrix and mounted-execution studies. Composition and adaptive discovery remain outside this analysis (Section 6).

B PROOF OF THE COVERAGE AND SINGLETON REDUNDANCY RESULT

The proof of Theorem 1 has two parts. We establish diminishing returns for each task and average over tasks to obtain the classical greedy guarantee. We then construct a pool in which singleton ranking selects redundant skills.

B.1 PRELIMINARIES AND ASSUMPTIONS

Fix the candidate pool $V = \{1, \dots, m\}$ and a separate baseline $0 \notin V$. A library $S \subseteq V$ satisfies $|S| \le k$, and tasks follow $Z \sim P$.

For task z and skill e, clip a pre-specified affine transform of the raw reward–cost score:

$$Y_e(z) = \mathrm{clip}(a\, R_e(z) + b,\ 0,\ 1)\,. \tag{3}$$

The task-conditional mean utility is $u_e(z) = \mathbb{E}[Y_e(z)]$.

Adding a skill does not alter any other skill's utility:

$$u_e(z; S) = u_e(z) \qquad \text{for all } e \in V \cup \{0\},\ z \in \mathcal{Z},\ S \subseteq V.$$

Under the one-skill-per-task oracle evaluation used by SkillCover, the task-level value of a library is

$$f_z(S) = \max_{e \in S \cup \{0\}} u_e(z) - u_0(z).$$

The population oracle objective is

$$F(S) = \mathbb{E}_{Z \sim P}[f_Z(S)]\,. \tag{4}$$

For a set function G, greedy starts from $S_0 = \emptyset$ and at iteration t adds the best remaining element,

$$e_t \in \arg\max_{e \in V \setminus S_t} [G(S_t \cup \{e\}) - G(S_t)]\,, \qquad S_{t+1} = S_t \cup \{e_t\},$$

and after k steps the greedy library is $S_g = S_k$.

We restate the assumptions needed for Theorem 1.

Assumption 4 (Fixed pool, baseline, and budget)**.** *The candidate pool, baseline, and budget are fixed before evaluation: $V = \{1, \dots, m\}$, $0 \notin V$, and $k \in \{1, \dots, m\}$.*

Assumption 5 (Bounded clipped utilities)**.** *The adjusted one-rollout utility is clipped before any coverage computation: $Y_e(z) = \mathrm{clip}(a\, R_e(z) + b,\ 0,\ 1)$, $Y_e(z) \in [0, 1]$, and $u_e(z) = \mathbb{E}[Y_e(z)] \in [0, 1]$.*

Assumption 6 (Atomic one-skill-per-task evaluation)**.** *Each task is served by at most one selected skill or the baseline, and adding one skill does not change another skill's utility:*

$$f_z(S) = \max_{e \in S \cup \{0\}} u_e(z) - u_0(z), \qquad u_e(z; S) = u_e(z).$$

Assumption 7 (Measurability and integrability)**.** *All displayed expectations are well defined: $\mathbb{E}[|Y_e(Z)|] < \infty$ and $\mathbb{E}[|f_Z(S)|] < \infty$.*

B.2 SUPPORTING LEMMAS

The first lemma establishes bounded normalization (Section 3).

Lemma 1 (Bounded normalization lemma)**.** *Assume the rollout utility is defined by the clipped affine map $Y_e(z) = \mathrm{clip}(a\, R_e(z) + b,\ 0,\ 1)$. Then, for every skill e and task z, $0 \le Y_e(z) \le 1$, and consequently $0 \le u_e(z) \le 1$. Moreover, if baseline-relative values are formed* after *clipping, then for every $S \subseteq V$ and $z \in \mathcal{Z}$,*

$$\Delta_{\mathrm{base}}(S, z) = \max_{e \in S \cup \{0\}} Y_e(z) - Y_0(z) \in [0, 1].$$

Proof. By the definition of the clipping operator, $0 \le \mathrm{clip}(a\,R_e(z)+b,\ 0,\ 1) \le 1$, hence $0 \le Y_e(z) \le 1$. Taking expectations preserves the interval, $0 \le \mathbb{E}[Y_e(z)] \le 1$, so $0 \le u_e(z) \le 1$. For the baseline-relative quantity formed after clipping, we have $Y_0(z) \le \max_{e\in S\cup\{0\}} Y_e(z) \le 1$, and subtracting $Y_0(z)$ gives

$$0 \le \max_{e\in S\cup\{0\}} Y_e(z) - Y_0(z) \le 1 - Y_0(z) \le 1.$$

Thus $\Delta_{\mathrm{base}}(S,z) \in [0,1]$. This is the bounded quantity used in the coverage objective. □

For example, suppose $a = \frac{1}{2}$, $b = \frac{1}{5}$, the baseline raw score is $R_0(z) = 0.1$, and a candidate raw score is $R_1(z) = 1.4$. Then

$$Y_0(z) = \mathrm{clip}\left(\frac{1}{2}\cdot 0.1 + \frac{1}{5}, 0, 1\right) = 0.25,$$
$$Y_1(z) = \mathrm{clip}\left(\frac{1}{2}\cdot 1.4 + \frac{1}{5}, 0, 1\right) = 0.90,$$

so the post-clipping baseline-relative gain is $\Delta_{\mathrm{base}}(\{1\}, z) = 0.90 - 0.25 = 0.65$.

The pointwise maximum gives the diminishing-returns property used by greedy selection.

Lemma 2 (Pointwise maximum diminishing-returns lemma)**.** *Fix a task $z \in \mathcal{Z}$. Define*

$$f_z(S) = \max_{e\in S\cup\{0\}} u_e(z) - u_0(z).$$

Then f_z is normalized, $f_z(\emptyset) = 0$, and monotone, $A \subseteq B \Longrightarrow f_z(A) \le f_z(B)$. It is submodular in the diminishing-returns sense:

$$A \subseteq B, \quad e \notin B \quad \Longrightarrow \quad f_z(A\cup\{e\}) - f_z(A) \ge f_z(B\cup\{e\}) - f_z(B).$$

Equivalently, the marginal gain $\Delta(e \mid S; z) = f_z(S\cup\{e\}) - f_z(S)$ satisfies

$$\Delta(e \mid S; z) = \max\left(0,\ u_e(z) - \max_{a\in S\cup\{0\}} u_a(z)\right).$$

Proof. Define the current best available utility on task z by $M_S(z) = \max_{a\in S\cup\{0\}} u_a(z)$, so that $f_z(S) = M_S(z) - u_0(z)$. Normalization follows from $f_z(\emptyset) = \max_{e\in\emptyset\cup\{0\}} u_e(z) - u_0(z) = u_0(z) - u_0(z) = 0$. Monotonicity follows because $A \subseteq B$ implies

$$M_A(z) = \max_{a\in A\cup\{0\}} u_a(z) \le \max_{a\in B\cup\{0\}} u_a(z) = M_B(z),$$
$$f_z(A) = M_A(z) - u_0(z) \le M_B(z) - u_0(z) = f_z(B).$$

For the marginal gain of adding $e \notin S$,

$$\begin{aligned}\Delta(e \mid S; z) &= f_z(S\cup\{e\}) - f_z(S)\\ &= \max\{M_S(z), u_e(z)\} - u_0(z) - (M_S(z) - u_0(z))\\ &= \max\{M_S(z), u_e(z)\} - M_S(z)\\ &= \max(0,\ u_e(z) - M_S(z))\,.\end{aligned}$$

Now let $A \subseteq B$, so that $M_A(z) \le M_B(z)$. Since the map $x \mapsto \max(0, u_e(z) - x)$ is nonincreasing in x,

$$\begin{aligned}\Delta(e \mid A; z) &= \max(0,\ u_e(z) - M_A(z))\\ &\ge \max(0,\ u_e(z) - M_B(z))\\ &= \Delta(e \mid B; z).\end{aligned}$$

This is the submodularity inequality. □

For example, suppose $u_0(z) = 0.2$, $u_1(z) = 0.7$, and $u_2(z) = 0.6$. The one-skill library value is $f_z(\{1\}) = 0.7 - 0.2 = 0.5$, and adding skill 2 leaves the maximum unchanged: $f_z(\{1,2\}) = 0.7 - 0.2 = 0.5$. Its marginal gain after skill 1 is therefore $\Delta(2 \mid \{1\}; z) = 0$.

These properties carry over to population coverage by taking expectations.

Proposition 1 (Expectation preserves normalization, monotonicity, and submodularity)**.** *Assume that for every task z, the set function $f_z : 2^V \to \mathbb{R}$ is normalized, monotone, and submodular. Define $F(S) = \mathbb{E}_{Z\sim P}[f_Z(S)]$. Then F is normalized, $F(\emptyset) = 0$, and monotone, $A \subseteq B \implies F(A) \leq F(B)$. It is submodular:*

$$A \subseteq B, \quad e \notin B \quad \implies \quad F(A \cup \{e\}) - F(A) \geq F(B \cup \{e\}) - F(B).$$

Proof. Normalization is $F(\emptyset) = \mathbb{E}[f_Z(\emptyset)] = \mathbb{E}[0] = 0$. For monotonicity, if $A \subseteq B$, then $f_Z(A) \leq f_Z(B)$ almost surely, so $F(A) = \mathbb{E}[f_Z(A)] \leq \mathbb{E}[f_Z(B)] = F(B)$. For submodularity, if $A \subseteq B$ and $e \notin B$, then

$$\begin{aligned} F(A \cup \{e\}) - F(A) &= \mathbb{E}[f_Z(A \cup \{e\}) - f_Z(A)] \\ &\geq \mathbb{E}[f_Z(B \cup \{e\}) - f_Z(B)] \\ &= F(B \cup \{e\}) - F(B). \end{aligned}$$

Thus F inherits all three properties. □

For example, suppose $\mathbb{P}(Z = z_1) = \frac{1}{2}$ and $\mathbb{P}(Z = z_2) = \frac{1}{2}$. If $f_{z_1}(\{1\}) = 1$ and $f_{z_2}(\{1\}) = 0$, then $F(\{1\}) = \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 0 = \frac{1}{2}$. The normalization, monotonicity and diminishing-returns inequalities average in the same way.

B.3 Greedy approximation and the clone construction

We restate Theorem 1 from Section 3.

Theorem 4 (Coverage, greedy approximation, and singleton redundancy)**.** *Assume the candidate pool $V = \{1, \ldots, m\}$ is fixed, the baseline $0 \notin V$ is available, and adjusted utilities are bounded, $u_e(Z) \in [0, 1]$ for all $e \in V \cup \{0\}$. Assume one-skill-per-task oracle evaluation and the population objective*

$$f_z(S) = \max_{e \in S \cup \{0\}} u_e(z) - u_0(z), \qquad F(S) = \mathbb{E}_{Z\sim P}[f_Z(S)].$$

Then F is normalized, $F(\emptyset) = 0$, and monotone, $A \subseteq B \implies F(A) \leq F(B)$. It is submodular:

$$A \subseteq B, \quad e \notin B \quad \implies \quad F(A \cup \{e\}) - F(A) \geq F(B \cup \{e\}) - F(B).$$

If exact greedy is run for k steps on F, then the returned library S_g satisfies

$$F(S_g) \geq \left(1 - \frac{1}{e}\right) \max_{|S| \leq k} F(S).$$

Moreover, for every $k \geq 2$ and every $\eta > 0$, there exists a binary coverage instance with $u_0(z) = 0$ for all z such that top-k singleton ranking satisfies

$$\frac{F(S_{\text{singleton}})}{F(S^\star)} \leq \frac{1}{k} + \eta,$$

where $S_{\text{singleton}} =$ top-k skills by $F(\{e\})$ and $S^\star \in \arg\max_{|S| \leq k} F(S)$.

Proof. The structural part follows directly from the preceding lemmas. By Lemma 1, $u_e(z) \in [0, 1]$. By Lemma 2, for every fixed task z, the taskwise set function $f_z(S) = \max_{e \in S \cup \{0\}} u_e(z) - u_0(z)$ is normalized, monotone, and submodular. By Proposition 1, the population objective $F(S) = \mathbb{E}[f_Z(S)]$ is also normalized, monotone, and submodular. This proves

$$\begin{aligned} F(\emptyset) &= 0, \\ A \subseteq B &\implies F(A) \leq F(B), \\ A \subseteq B,\ e \notin B &\implies F(A \cup \{e\}) - F(A) \geq F(B \cup \{e\}) - F(B). \end{aligned}$$

For exact greedy under the cardinality budget, the classical Nemhauser–Wolsey–Fisher theorem for monotone submodular maximization gives

$$F(S_g) \geq \left(1 - \frac{1}{e}\right) \max_{|S| \leq k} F(S).$$

We import this result from Nemhauser et al. (1978) rather than re-prove it here.

It remains to prove the singleton-ranking separation. Fix any $k \geq 2$ and any $\eta > 0$, and choose a small positive number $\varepsilon \in (0, \eta)$. Define the ratio parameter $r = 1 + \varepsilon$, the diverse-region mass $q = \frac{1}{2(r+k-1)}$, and the clone-region mass $p = rq$. Then $p > q$, and

$$p + (k-1)q = rq + (k-1)q = (r+k-1)q = \frac{1}{2},$$

so these masses fit inside a probability space.

Partition the task space into disjoint regions $C,\ D_1,\ \ldots,\ D_{k-1},\ E$, with probabilities $\mathbb{P}(C) = p$, $\mathbb{P}(D_j) = q$ for $j = 1, \ldots, k-1$, and $\mathbb{P}(E) = 1 - p - (k-1)q = \frac{1}{2}$.

Create k clone skills $c_1, \ldots, c_k$ and $k-1$ diverse skills $d_1, \ldots, d_{k-1}$. Set the baseline utility to zero everywhere, $u_0(z) = 0$ for all z, and define clone and diverse utilities by

$$u_{c_j}(z) = \mathbf{1}\{z \in C\} \quad \text{for } j = 1, \ldots, k, \qquad u_{d_j}(z) = \mathbf{1}\{z \in D_j\} \quad \text{for } j = 1, \ldots, k-1.$$

For every clone c_j and every diverse skill d_j, the singleton values are

$$F(\{c_j\}) = \mathbb{E}\big[\max\{u_{c_j}(Z), u_0(Z)\} - u_0(Z)\big] = \mathbb{E}[u_{c_j}(Z)] = \mathbb{P}(Z \in C) = p,$$
$$F(\{d_j\}) = \mathbb{E}[u_{d_j}(Z)] = \mathbb{P}(Z \in D_j) = q.$$

Since $p > q$, the top-k singleton ranking is forced to choose the k clones, $S_{\text{singleton}} = \{c_1, \ldots, c_k\}$, whose library value is

$$F(S_{\text{singleton}}) = \mathbb{E}\left[\max_{e \in \{c_1, \ldots, c_k\}} u_e(Z)\right] = \mathbb{P}(Z \in C) = p.$$

Now consider the library $S^\star = \{c_1, d_1, \ldots, d_{k-1}\}$. Its value is

$$F(S^\star) = \mathbb{E}\left[\max_{e \in S^\star} u_e(Z)\right] = \mathbb{P}(Z \in C \cup D_1 \cup \cdots \cup D_{k-1}) = p + (k-1)q.$$

No size-k library can exceed this value because the regions $C, D_1, \ldots, D_{k-1}$ are disjoint and these k skills already cover all of them. Hence $S^\star$ is optimal. Therefore, substituting $p = rq$,

$$\frac{F(S_{\text{singleton}})}{F(S^\star)} = \frac{p}{p + (k-1)q} = \frac{r}{r+k-1}.$$

We compare this to $1/k$:

$$\begin{aligned}
\frac{r}{r+k-1} - \frac{1}{k} &= \frac{kr - (r+k-1)}{k(r+k-1)} \\
&= \frac{(k-1)(r-1)}{k(r+k-1)} \\
&\leq r - 1 \\
&= \varepsilon \\
&< \eta.
\end{aligned}$$

Hence $\frac{F(S_{\text{singleton}})}{F(S^\star)} \leq \frac{1}{k} + \eta$, which proves the claimed worst-case redundancy of singleton ranking. □

For $k = 3$, take clone-region mass $p = 0.17$ and diverse-region mass $q = 0.16$. Singleton ranking picks three clones, giving $F(S_{\text{singleton}}) = p = 0.17$. The optimal size-3 library picks one clone and two diverse skills, giving $F(S^\star) = p + 2q = 0.49$. The ratio is

$$\frac{F(S_{\text{singleton}})}{F(S^\star)} = \frac{0.17}{0.49} \approx 0.347,$$

close to $\frac{1}{3} \approx 0.333$.

The redundant-clone synthetic study in Section 4 uses the construction from Theorem 1. It measures the ratio $\frac{F(S_{\text{singleton}})}{F(S^\star)} = \frac{p}{p+(k-1)q}$, with a worst observed value at $k = 5$ of 0.2052, close to the limiting benchmark $\frac{1}{5} = 0.2$. The experiment illustrates the separation; it is not part of the proof.

The result controls the oracle library value

$$F(S_g) = \mathbb{E}\left[\max_{e \in S_g \cup \{0\}} u_e(Z) - u_0(Z)\right];$$

which need not equal routed value $J_v(\rho, S_g)$. Appendix D treats routing under additional assumptions.

C Proof of the guarantee under noisy calibration

To prove Theorem 2, we bound task-sampling error over all libraries of size at most k and rollout error in every calibration cell. Stability of the pointwise maximum converts cell error into objective error. The greedy transfer lemma then gives a population guarantee for the selected library.

C.1 Uniform concentration over libraries and cells

Lemma 3 (Exponential bound on the number of size-k libraries)**.** *Assume $1 \le k \le m$ and define $N_k = \sum_{j=0}^{k} \binom{m}{j}$. Then*

$$N_k \le \left(\frac{em}{k}\right)^k.$$

Proof. Set $p = \frac{k}{m}$. Because $0 < p \le 1$, we have $p^j \ge p^k$ for every $0 \le j \le k$. Therefore

$$(1+p)^m = \sum_{j=0}^{m} \binom{m}{j} p^j \ge \sum_{j=0}^{k} \binom{m}{j} p^j \ge p^k \sum_{j=0}^{k} \binom{m}{j} = p^k N_k,$$

$$N_k \le p^{-k}(1+p)^m = \left(\frac{m}{k}\right)^k \left(1 + \frac{k}{m}\right)^m \le \left(\frac{m}{k}\right)^k e^k = \left(\frac{em}{k}\right)^k.$$

This proves the claim. □

For $m = 10$ and $k = 2$, $N_2 = \binom{10}{0} + \binom{10}{1} + \binom{10}{2} = 56$, while $\left(\frac{em}{k}\right)^k = (5e)^2 \approx 184.7$.

Lemma 4 (Finite-library uniform convergence)**.** *Assume the calibration tasks are i.i.d., $Z_1, \dots, Z_n \overset{\text{iid}}{\sim} P$. Define the taskwise oracle coverage by $f_z(S) = \max_{e \in S \cup \{0\}} u_e(z) - u_0(z)$, and assume bounded utilities: $u_e(z) \in [0,1]$ for all $e \in V \cup \{0\}$, $z \in \mathcal{Z}$. Define the population objective $F(S) = \mathbb{E}[f_Z(S)]$, the noiseless empirical objective $F_n(S) = \frac{1}{n}\sum_{i=1}^{n} f_{Z_i}(S)$, and $\beta_n = \sqrt{\frac{\log(4N_k/\delta)}{2n}}$. Then, with probability at least $1 - \frac{\delta}{2}$, the uniform deviation event*

$$\sup_{|S| \le k} |F_n(S) - F(S)| \le \beta_n$$

holds.

Proof. Fix any library satisfying $|S| \le k$. Because $u_e(Z) \in [0,1]$ for every $e \in V \cup \{0\}$, we have $0 \le f_{Z_i}(S) \le 1$ for every i, and $\mathbb{E}[f_{Z_i}(S)] = F(S)$. Hoeffding's inequality therefore gives

$$\Pr(|F_n(S) - F(S)| > \beta) \le 2\exp(-2n\beta^2) \qquad \text{for every } \beta > 0.$$

Applying a union bound over all libraries with $|S| \le k$ yields

$$\begin{aligned}\Pr\left(\sup_{|S| \le k} |F_n(S) - F(S)| > \beta\right) &\le \sum_{|S| \le k} \Pr(|F_n(S) - F(S)| > \beta) \\ &\le 2N_k \exp(-2n\beta^2).\end{aligned}$$

Choosing $\beta = \beta_n = \sqrt{\frac{\log(4N_k/\delta)}{2n}}$ makes the right-hand side

$$2N_k \exp(-2n\beta_n^2) = 2N_k \exp\left(-\log\frac{4N_k}{\delta}\right) = 2N_k \cdot \frac{\delta}{4N_k} = \frac{\delta}{2},$$

so $\Pr\left(\sup_{|S|\le k} |F_n(S) - F(S)| \le \beta_n\right) \ge 1 - \frac{\delta}{2}$. □

For example, if $n = 2$, $f_{Z_1}(S) = 0.6$, and $f_{Z_2}(S) = 0.2$, then $F_n(S) = \frac{0.6+0.2}{2} = 0.4$. On the event in the lemma, each empirical average is within β_n of its population mean $F(S)$, simultaneously for every $|S| \le k$.

Proposition 2 (Coordinate concentration for rollout means)**.** *Assume the rollout mean in calibration cell (i, e) is $\hat{u}_{i,e} = \frac{1}{q}\sum_{r=1}^{q} Y_{i,e,r}$, conditionally unbiased, $\mathbb{E}[\hat{u}_{i,e} \mid Z_i] = u_e(Z_i)$, and conditionally sub-Gaussian: $\hat{u}_{i,e} - u_e(Z_i)$ is conditionally $\frac{\sigma}{\sqrt{q}}$-sub-Gaussian. Define $\beta_q = \sigma\sqrt{\frac{2\log(4(m+1)n/\delta)}{q}}$. Then, with probability at least $1 - \frac{\delta}{2}$, the entrywise event*

$$\max_{1\le i\le n,\ e\in V\cup\{0\}} |\hat{u}_{i,e} - u_e(Z_i)| \le \beta_q$$

holds.

Proof. Fix any $i \in \{1, \dots, n\}$ and any $e \in V \cup \{0\}$. By the conditional sub-Gaussian assumption,

$$\Pr(|\hat{u}_{i,e} - u_e(Z_i)| > \beta \mid Z_i) \le 2\exp\left(-\frac{q\beta^2}{2\sigma^2}\right) \qquad \text{for every } \beta > 0,$$

and taking expectation over Z_i preserves the same bound: $\Pr(|\hat{u}_{i,e} - u_e(Z_i)| > \beta) \le 2\exp\left(-\frac{q\beta^2}{2\sigma^2}\right)$. There are exactly $(m+1)n$ task-skill cells. Another union bound gives

$$\Pr\left(\max_{1\le i\le n,\ e\in V\cup\{0\}} |\hat{u}_{i,e} - u_e(Z_i)| > \beta\right) \le 2(m+1)n\exp\left(-\frac{q\beta^2}{2\sigma^2}\right).$$

Choosing $\beta = \beta_q = \sigma\sqrt{\frac{2\log(4(m+1)n/\delta)}{q}}$ makes the right-hand side

$$\begin{aligned} 2(m+1)n\exp\left(-\frac{q\beta_q^2}{2\sigma^2}\right) &= 2(m+1)n\exp\left(-\log\frac{4(m+1)n}{\delta}\right) \\ &= 2(m+1)n \cdot \frac{\delta}{4(m+1)n} = \frac{\delta}{2}. \end{aligned}$$

This proves the claim. □

For example, if $\sigma = \frac{1}{2}$, $m = 9$, $n = 100$, $q = 25$, and $\delta = 0.1$, then $\beta_q = \frac{1}{2}\sqrt{\frac{2\log(4\cdot 10\cdot 100/0.1)}{25}}$. All $(m+1)n = 1000$ entries are simultaneously within β_q of their conditional means with probability at least 0.95.

C.2 Stability of coverage and transfer of the greedy guarantee

Lemma 5 (Max-stability of baseline-relative coverage)**.** *Fix any calibration tasks $Z_1, \dots, Z_n$, and assume the entrywise perturbation event*

$$\max_{1\le i\le n,\ e\in V\cup\{0\}} |\hat{u}_{i,e} - u_e(Z_i)| \le \alpha$$

holds for some $\alpha \ge 0$. Then, for every task index i and every library S,

$$\left|\max_{e\in S\cup\{0\}} \hat{u}_{i,e} - \max_{e\in S\cup\{0\}} u_e(Z_i)\right| \le \alpha.$$

Consequently,

$$\sup_{|S|\le k} \left|\hat{F}_n(S) - F_n(S)\right| \le 2\alpha,$$

where $\hat{F}_n(S) = \frac{1}{n}\sum_{i=1}^{n}\left[\max_{e\in S\cup\{0\}} \hat{u}_{i,e} - \hat{u}_{i,0}\right]$.

Proof. Fix any $i \in \{1, \dots, n\}$ and any library $S \subseteq V$. For every $e \in S \cup \{0\}$, the assumed event gives $u_e(Z_i) - \alpha \le \hat{u}_{i,e} \le u_e(Z_i) + \alpha$. Taking maxima over $e \in S \cup \{0\}$ yields

$$\max_{e \in S \cup \{0\}} u_e(Z_i) - \alpha \le \max_{e \in S \cup \{0\}} \hat{u}_{i,e} \le \max_{e \in S \cup \{0\}} u_e(Z_i) + \alpha,$$

so $\left|\max_{e \in S \cup \{0\}} \hat{u}_{i,e} - \max_{e \in S \cup \{0\}} u_e(Z_i)\right| \le \alpha$. The baseline coordinate also satisfies $|\hat{u}_{i,0} - u_0(Z_i)| \le \alpha$. Subtracting the two terms and using the triangle inequality gives

$$\begin{aligned}
&\left| \left[\max_{e \in S \cup \{0\}} \hat{u}_{i,e} - \hat{u}_{i,0} \right] - \left[\max_{e \in S \cup \{0\}} u_e(Z_i) - u_0(Z_i) \right] \right| \\
&\le \left| \max_{e \in S \cup \{0\}} \hat{u}_{i,e} - \max_{e \in S \cup \{0\}} u_e(Z_i) \right| + |\hat{u}_{i,0} - u_0(Z_i)| \\
&\le \alpha + \alpha \\
&= 2\alpha.
\end{aligned}$$

Averaging over i preserves the same bound:

$$\begin{aligned}
\left|\hat{F}_n(S) - F_n(S)\right| &= \left| \frac{1}{n} \sum_{i=1}^{n} \Big([\max_{e \in S \cup \{0\}} \hat{u}_{i,e} - \hat{u}_{i,0}] - [\max_{e \in S \cup \{0\}} u_e(Z_i) - u_0(Z_i)] \Big) \right| \\
&\le \frac{1}{n} \sum_{i=1}^{n} 2\alpha \\
&= 2\alpha.
\end{aligned}$$

Since the right-hand side does not depend on S, the same inequality holds uniformly over all $|S| \le k$. □

The factor 2α accounts for changes in both the maximum and the baseline. Suppose for one task $\max_{e \in S \cup \{0\}} u_e(Z_i) = 0.70$, $u_0(Z_i) = 0.20$, and $\alpha = 0.05$. If the maximum increases to 0.74 and the baseline decreases to 0.17, coverage changes from $0.70 - 0.20 = 0.50$ to $0.74 - 0.17 = 0.57$, a deviation of $0.07 \le 2\alpha = 0.10$.

Lemma 6 (Greedy transfer)**.** *Let $G : 2^V \to \mathbb{R}$ and $H : 2^V \to \mathbb{R}$ be normalized, monotone, and submodular. Assume $\sup_{|S| \le k} |G(S) - H(S)| \le \varepsilon$, and let $a_g = 1 - \frac{1}{e}$. Let S_H^g be the k-step exact greedy solution for H. Then*

$$H(S_H^g) \ge a_g \max_{|S| \le k} H(S) \qquad \text{and} \qquad G(S_H^g) \ge a_g \max_{|S| \le k} G(S) - (1 + a_g)\varepsilon;$$

equivalently, the second bound reads $G(S_H^g) \ge \left(1 - \frac{1}{e}\right) \max_{|S| \le k} G(S) - \left(2 - \frac{1}{e}\right) \varepsilon$.

Proof. The greedy approximation for monotone submodular maximization gives $H(S_H^g) \ge a_g \max_{|S| \le k} H(S)$. Let $S_G^\star \in \arg\max_{|S| \le k} G(S)$. Then

$$\begin{aligned}
G(S_H^g) &\ge H(S_H^g) - \varepsilon \\
&\ge a_g \max_{|S| \le k} H(S) - \varepsilon \\
&\ge a_g H(S_G^\star) - \varepsilon \\
&\ge a_g \big(G(S_G^\star) - \varepsilon \big) - \varepsilon \\
&= a_g \max_{|S| \le k} G(S) - (1 + a_g)\varepsilon.
\end{aligned}$$

Substituting $a_g = 1 - \frac{1}{e}$ yields $1 + a_g = 2 - \frac{1}{e}$. This proves the displayed form. □

If $\varepsilon = 0.03$ and $\max_{|S| \le k} G(S) = 0.50$, then $a_g \max_{|S| \le k} G(S) = \left(1 - \frac{1}{e}\right) 0.50 \approx 0.316$. The transfer penalty is $(1 + a_g)\varepsilon = \left(2 - \frac{1}{e}\right) 0.03 \approx 0.049$, so the lemma guarantees $G(S_H^g) \ge 0.316 - 0.049 \approx 0.267$.

C.3 POPULATION COVERAGE GUARANTEE

We restate Theorem 2 from Section 3.

Theorem 5 (Population coverage under noisy calibration)**.** *Assume the candidate pool is fixed, $V = \{1, \dots, m\}$, the baseline label is fixed, $0 \notin V$, and the budget is fixed, $|S| \le k$. Assume the candidate pool, baseline protocol, utility definition, clipping constants, cost weights, descriptors, and deterministic greedy tie-breaking are fixed independently of the calibration outcomes.*

Assume the calibration tasks are i.i.d., $Z_1, \dots, Z_n \overset{\text{iid}}{\sim} P$, and that the clipped rollout utilities satisfy $Y_{i,e,r} \in [0,1]$ for all i, $e \in V \cup \{0\}$, $r \in \{1, \dots, q\}$. Assume the rollout mean in cell (i,e) is $\hat{u}_{i,e} = \frac{1}{q}\sum_{r=1}^{q} Y_{i,e,r}$, conditionally unbiased, $\mathbb{E}[\hat{u}_{i,e} \mid Z_i] = u_e(Z_i)$, and conditionally sub-Gaussian: $\hat{u}_{i,e} - u_e(Z_i)$ is conditionally $\frac{\sigma}{\sqrt{q}}$-sub-Gaussian. Define the population oracle value, the noiseless empirical objective, and the realized noisy empirical objective by

$$F(S) = \mathbb{E}\left[\max_{e \in S \cup \{0\}} u_e(Z) - u_0(Z)\right],$$

$$F_n(S) = \frac{1}{n}\sum_{i=1}^{n}\left[\max_{e \in S \cup \{0\}} u_e(Z_i) - u_0(Z_i)\right], \qquad \hat{F}_n(S) = \frac{1}{n}\sum_{i=1}^{n}\left[\max_{e \in S \cup \{0\}} \hat{u}_{i,e} - \hat{u}_{i,0}\right].$$

Define the library count $N_k = \sum_{j=0}^{k}\binom{m}{j}$, the greedy constant $a_g = 1 - \frac{1}{e}$, and the task-sampling and rollout-noise terms

$$\beta_n = \sqrt{\frac{\log(4N_k/\delta)}{2n}}, \qquad \beta_q = \sigma\sqrt{\frac{2\log(4(m+1)n/\delta)}{q}}.$$

Let $\hat{S}_g$ be the k-step exact greedy solution for $\hat{F}_n$: $\hat{S}_g = S_k$. Then, with probability at least $1 - \delta$, the uniform deviation bound and the population guarantee

$$\sup_{|S| \le k}\left|\hat{F}_n(S) - F(S)\right| \le \beta_n + 2\beta_q,$$

$$F(\hat{S}_g) \ge a_g \max_{|S| \le k} F(S) - (1 + a_g)(\beta_n + 2\beta_q)$$

hold simultaneously.

Proof. We introduce two high-probability events, the task-sampling event and the rollout-estimation event:

$$\mathcal{E}_n = \left\{\sup_{|S| \le k} |F_n(S) - F(S)| \le \beta_n\right\}, \qquad \mathcal{E}_q = \left\{\max_{1 \le i \le n,\ e \in V \cup \{0\}} |\hat{u}_{i,e} - u_e(Z_i)| \le \beta_q\right\}.$$

By Lemma 4, $\Pr(\mathcal{E}_n) \ge 1 - \frac{\delta}{2}$, and by Proposition 2, $\Pr(\mathcal{E}_q) \ge 1 - \frac{\delta}{2}$; therefore, by another union bound, $\Pr(\mathcal{E}_n \cap \mathcal{E}_q) \ge 1 - \delta$.

We now work on the event $\mathcal{E} = \mathcal{E}_n \cap \mathcal{E}_q$. Lemma 5 applied with $\alpha = \beta_q$ gives $\sup_{|S| \le k}\left|\hat{F}_n(S) - F_n(S)\right| \le 2\beta_q$. Hence, for every library S with $|S| \le k$,

$$\left|\hat{F}_n(S) - F(S)\right| \le \left|\hat{F}_n(S) - F_n(S)\right| + |F_n(S) - F(S)| \le 2\beta_q + \beta_n,$$

and taking the supremum over $|S| \le k$ yields $\sup_{|S| \le k}\left|\hat{F}_n(S) - F(S)\right| \le \beta_n + 2\beta_q$. This proves the first claim.

For the greedy part, fix the realized calibration matrix $\{\hat{u}_{i,e}\}_{1 \le i \le n,\ e \in V \cup \{0\}}$, and for each task i define the deterministic taskwise set function $\hat{f}_i(S) = \max_{e \in S \cup \{0\}} \hat{u}_{i,e} - \hat{u}_{i,0}$. By the same pointwise-max diminishing-returns argument used earlier for Lemma 2, each $\hat{f}_i$ is normalized, monotone, and submodular. Averaging over i preserves these properties, so $\hat{F}_n(S) = \frac{1}{n}\sum_{i=1}^{n}\hat{f}_i(S)$ is also normalized, monotone, and submodular. Therefore exact greedy under the cardinality budget satisfies $\hat{F}_n(\hat{S}_g) \ge a_g \max_{|S| \le k} \hat{F}_n(S)$.

We now transfer this empirical guarantee to the population objective. On the event $\mathcal{E}$, the uniform closeness condition $\sup_{|S|\le k}\left|F(S)-\hat{F}_n(S)\right| \le \beta_n + 2\beta_q$ holds. Applying Lemma 6 with $G = F$, $H = \hat{F}_n$, and $\varepsilon = \beta_n + 2\beta_q$, we obtain

$$F(\hat{S}_g) \ge a_g \max_{|S|\le k} F(S) - (1 + a_g)(\beta_n + 2\beta_q).$$

Since all steps hold on $\mathcal{E}$ and $\Pr(\mathcal{E}) \ge 1 - \delta$, the theorem follows. □

For example, suppose $\max_{|S|\le k} F(S) = 0.60$ and the error bound is $\beta_n + 2\beta_q = 0.05$. The theorem gives

$$F(\hat{S}_g) \ge \left(1 - \frac{1}{e}\right) 0.60 - \left(2 - \frac{1}{e}\right) 0.05 \approx 0.379 - 0.082 \approx 0.297.$$

The numerical lower bound can be conservative. Section 4 compares the qualitative n- and q-scaling trends with synthetic results.

Corollary 1 (Asymptotic form of the excess term)**.** *Under the assumptions of Theorem 2,*

$$F(\hat{S}_g) \ge \left(1 - \frac{1}{e}\right) \max_{|S|\le k} F(S) - O\left(\sqrt{\frac{k\log(em/k) + \log(1/\delta)}{n}} + \sigma\sqrt{\frac{\log((m+1)n/\delta)}{q}}\right).$$

Proof. By Lemma 3, $N_k \le \left(\frac{em}{k}\right)^k$. Therefore

$$\beta_n = \sqrt{\frac{\log(4N_k/\delta)}{2n}} \le \sqrt{\frac{\log(4/\delta) + k\log(em/k)}{2n}},$$
$$\beta_q = \sigma\sqrt{\frac{2\log(4(m+1)n/\delta)}{q}} = O\left(\sigma\sqrt{\frac{\log((m+1)n/\delta)}{q}}\right).$$

Substituting these bounds into Theorem 2 yields the result. □

Corollary 2 (Hoeffding instantiation)**.** *Under the assumptions of Theorem 2, if, conditional on each task Z_i, the rollouts within each cell are independent with common mean $u_e(Z_i)$ and satisfy $Y_{i,e,r} \in [0,1]$ almost surely, then the sub-Gaussian parameter may be taken as $\sigma = \frac{1}{2}$. Consequently,*

$$\beta_q = \frac{1}{2}\sqrt{\frac{2\log(4(m+1)n/\delta)}{q}}.$$

Proof. For each fixed i and e, the conditional support of the centered rollout variable has length one because $Y_{i,e,r} \in [0,1]$. Hoeffding's lemma therefore implies conditional $1/2$-sub-Gaussianity for the centered one-rollout variable. Averaging q independent rollouts divides the proxy variance by q, so $\hat{u}_{i,e} - u_e(Z_i)$ is conditionally $\frac{1}{2\sqrt{q}}$-sub-Gaussian. This is exactly the theorem's assumption with $\sigma = \frac{1}{2}$. Substituting into the definition of β_q gives the final display. □

This result concerns population oracle coverage $F(\hat{S}_g) = \mathbb{E}\left[\max_{e\in\hat{S}_g\cup\{0\}} u_e(Z) - u_0(Z)\right]$. Routed utility $J_v(\rho, \hat{S}_g)$ is treated next. The synthetic n- and q-sweeps in Section 4 check the two error terms qualitatively; one path uses Gaussian cell noise in place of repeated rollouts.

D PROOF OF THE ROUTING GUARANTEE AND SAMPLE SIZE CONSEQUENCES

We now prove the deployment theorem from Section 3. Theorem 2 bounds the population oracle value $F(S)$ available in a selected library. Here we bound the descriptor-conditional routed value $J_v(\rho, S)$ achieved when a router chooses a skill from the task descriptor, where

$$F(S) = \mathbb{E}\left[\max_{e\in S\cup\{0\}} u_e(Z) - u_0(Z)\right] \qquad \text{and} \qquad J_v(\rho, S) = \mathbb{E}\big[v_{\rho(X)}(X) - v_0(X)\big].$$

The library remains fixed, and the router invokes at most one skill per task.

D.1 NEAREST-WINNER ROUTING

Theorem 6 (Nearest-winner deployment is conditionally close to descriptor-conditional oracle)**.** *Fix a library $S \subseteq V$, and assume the deployment descriptor X takes values in a metric space $(\mathcal{X}, d)$. For each skill $e \in S \cup \{0\}$, define the descriptor-conditional expected adjusted utility by $v_e(x) = \mathbb{E}[Y_e \mid X = x]$, and define the descriptor-conditional oracle coverage value by $F_v(S) = \mathbb{E}\big[\max_{e\in S\cup\{0\}} v_e(X) - v_0(X)\big]$. Assume every v_e is L-Lipschitz on the evaluated descriptor support, i.e. $|v_e(x) - v_e(x')| \le L\, d(x, x')$ for all x, x' and all $e \in S \cup \{0\}$. Let the calibration descriptors be $x_1, \dots, x_n$, and assume the calibration winner estimates satisfy $\max_{1\le i\le n,\ e\in S\cup\{0\}} |\hat{u}_{i,e} - v_e(x_i)| \le \epsilon_{\text{desc}}$. For a test descriptor x, define the nearest calibration index by $i(x) \in \arg\min_{1\le i\le n} d(x, x_i)$ and the nearest-winner router by $\rho_S(x) \in \arg\max_{e\in S\cup\{0\}} \hat{u}_{i(x),e}$. Then, for every test descriptor x,*

$$v_{\rho_S(x)}(x) \ge \max_{e\in S\cup\{0\}} v_e(x) - 2L\, d(x, x_{i(x)}) - 2\epsilon_{\text{desc}},$$

$$\textit{and consequently} \qquad J_v(\rho_S, S) \ge F_v(S) - 2L\,\mathbb{E}\left[\min_{1\le i\le n} d(X, x_i)\right] - 2\epsilon_{\text{desc}}.$$

If the descriptor X is insufficient for the task Z, then the theorem applies only to the descriptor-conditional oracle quantity $F_v(S)$, not to the task-level oracle quantity $F(S)$.

Proof. Fix an arbitrary test descriptor $x \in \mathcal{X}$, write $i = i(x)$ for the nearest calibration index and $\hat{e} = \rho_S(x)$ for the routed action, and let $e^\star(x) \in \arg\max_{e\in S\cup\{0\}} v_e(x)$ be the descriptor-conditional oracle winner at x. Using, in order, Lipschitzness of $v_{\hat{e}}$, the calibration error assumption, the definition of $\hat{e}$ as an empirical winner at x_i, the calibration error assumption again, and Lipschitzness of $v_{e^\star(x)}$, we obtain

$$\begin{aligned}
v_{\rho_S(x)}(x) &= v_{\hat{e}}(x) \\
&\ge v_{\hat{e}}(x_i) - L\, d(x, x_i) \\
&\ge \hat{u}_{i,\hat{e}} - \epsilon_{\text{desc}} - L\, d(x, x_i) \\
&\ge \hat{u}_{i,e^\star(x)} - \epsilon_{\text{desc}} - L\, d(x, x_i) \\
&\ge v_{e^\star(x)}(x_i) - 2\epsilon_{\text{desc}} - L\, d(x, x_i) \\
&\ge v_{e^\star(x)}(x) - 2\epsilon_{\text{desc}} - 2L\, d(x, x_i) = \max_{e\in S\cup\{0\}} v_e(x) - 2L\, d(x, x_i) - 2\epsilon_{\text{desc}},
\end{aligned}$$

where the final equality uses $v_{e^\star(x)}(x) = \max_{e\in S\cup\{0\}} v_e(x)$. This is the pointwise bound.

To pass to the baseline-relative deployed value, subtract $v_0(x)$ from both sides:

$$v_{\rho_S(x)}(x) - v_0(x) \ge \max_{e\in S\cup\{0\}} v_e(x) - v_0(x) - 2L\, d(x, x_i) - 2\epsilon_{\text{desc}}.$$

Now take expectation over the test descriptor X:

$$\begin{aligned}
J_v(\rho_S, S) &= \mathbb{E}\big[v_{\rho_S(X)}(X) - v_0(X)\big] \\
&\ge \mathbb{E}\left[\max_{e\in S\cup\{0\}} v_e(X) - v_0(X)\right] - 2L\,\mathbb{E}\big[d(X, x_{i(X)})\big] - 2\epsilon_{\text{desc}} \\
&= F_v(S) - 2L\,\mathbb{E}\left[\min_{1\le i\le n} d(X, x_i)\right] - 2\epsilon_{\text{desc}}.
\end{aligned}$$

This proves the theorem. □

Suppose the descriptor-conditional oracle value at a test point is $\max_{e\in S\cup\{0\}} v_e(x) = 0.80$, the Lipschitz constant is $L = 2$, the nearest calibration descriptor is at distance $d(x, x_{i(x)}) = 0.05$, and the estimation error is $\epsilon_{\text{desc}} = 0.03$. The theorem then guarantees

$$v_{\rho_S(x)}(x) \geq 0.80 - 2(2)(0.05) - 2(0.03) = 0.80 - 0.20 - 0.06 = 0.54.$$

If the baseline value is $v_0(x) = 0.50$, the routed baseline-relative value satisfies $v_{\rho_S(x)}(x) - v_0(x) \geq 0.04$.

Section 4 examines routing with clean and corrupted descriptors. The reported regrets are $\text{Regret}_{\text{clean desc}} = 0.013$, $\text{Regret}_{\text{random embed}} = 0.178$, $\text{Regret}_{\text{missing variable}} = 0.135$, and $\text{Regret}_{\text{aliased}} = 0.074$. These diagnostics show sensitivity to descriptor quality; they do not separately estimate the two error terms in the theorem.

D.2 Baseline comparison and sample-size requirements

Corollary 3 (Loss relative to the baseline)**.** *Under the assumptions of Theorem 3, the routed skill never falls more than the theorem error below the baseline. For every test descriptor x,*

$$v_{\rho_S(x)}(x) - v_0(x) \geq -2L\, d(x, x_{i(x)}) - 2\epsilon_{\text{desc}},$$

$$\textit{and consequently} \qquad J_v(\rho_S, S) \geq -2L\, \mathbb{E}\left[\min_{1\leq i\leq n} d(X, x_i)\right] - 2\epsilon_{\text{desc}}.$$

Proof. Because the baseline is included in the maximization set, $\max_{e\in S\cup\{0\}} v_e(x) \geq v_0(x)$. Substituting this inequality into the pointwise bound of Theorem 3 gives

$$v_{\rho_S(x)}(x) \geq v_0(x) - 2L\, d(x, x_{i(x)}) - 2\epsilon_{\text{desc}}.$$

Subtracting $v_0(x)$ yields the first bound. Taking expectation yields the second. □

Corollary 3 bounds how far routed value can fall below the baseline. The deviation can be negative in finite samples, with magnitude controlled by descriptor distance and calibration error.

Corollary 4 (Coverage preservation consequence)**.** *Under the assumptions of Theorem 3, if a fixed library has descriptor-conditional oracle coverage level $F_v(S) \geq \gamma$, then its routed deployed value satisfies*

$$J_v(\rho_S, S) \geq \gamma - 2L\, \mathbb{E}\left[\min_{1\leq i\leq n} d(X, x_i)\right] - 2\epsilon_{\text{desc}}.$$

Proof. Insert the lower bound $F_v(S) \geq \gamma$ into the final inequality of Theorem 3. □

For a fixed library, concentration of the rollout means makes the dependence on q explicit.

Proposition 3 (Rollout-complexity consequence for a fixed deployed library)**.** *Fix a library $S \subseteq V$ and assume $|S| \leq k$. For every calibration descriptor x_i and every $e \in S \cup \{0\}$, suppose the estimate is formed from repeated rollouts, $\hat{u}_{i,e} = \frac{1}{q}\sum_{r=1}^{q} Y_{i,e,r}$, with descriptor-conditional cell mean $\mu_{i,e} = \mathbb{E}[\hat{u}_{i,e} \mid x_i]$. Assume the residual descriptor-aliasing error is bounded by $\max_{1\leq i\leq n,\, e\in S\cup\{0\}} |\mu_{i,e} - v_e(x_i)| \leq \eta_{\text{alias}}$, and assume the centered rollout average $\hat{u}_{i,e} - \mu_{i,e}$ is conditionally $\frac{\sigma}{\sqrt{q}}$-sub-Gaussian. Define the fixed-library rollout term by $\beta_q^{(S)} = \sigma\sqrt{\frac{2\log(4(k+1)n/\delta)}{q}}$. Then, with probability at least $1-\delta$, the descriptor estimation level satisfies $\epsilon_{\text{desc}} \leq \eta_{\text{alias}} + \beta_q^{(S)}$. On the same event,*

$$J_v(\rho_S, S) \geq F_v(S) - 2L\, \mathbb{E}\left[\min_{1\leq i\leq n} d(X, x_i)\right] - 2\eta_{\text{alias}} - 2\beta_q^{(S)}.$$

Therefore, if a target deployment regret level τ is desired and $\tau > 2L\, \mathbb{E}[\min_{1\leq i\leq n} d(X, x_i)] + 2\eta_{\text{alias}}$, then it is sufficient that

$$q \geq \frac{8\sigma^2 \log(4(k+1)n/\delta)}{\left(\tau - 2L\, \mathbb{E}[\min_{1\leq i\leq n} d(X, x_i)] - 2\eta_{\text{alias}}\right)^2}.$$

Proof. For any fixed calibration index i and skill $e \in S \cup \{0\}$, the sub-Gaussian assumption and the definition of $\beta_q^{(S)}$ give

$$\Pr\Big(|\hat{u}_{i,e} - \mu_{i,e}| > \beta_q^{(S)} \mid x_i\Big) \le 2\exp\left(-\frac{q(\beta_q^{(S)})^2}{2\sigma^2}\right) = 2\exp\left(-\log\left(\frac{4(k+1)n}{\delta}\right)\right) = \frac{\delta}{2(k+1)n}.$$

There are at most $(k+1)n$ relevant task-skill cells because the deployed library contains at most k skills plus the baseline. A union bound therefore yields

$$\max_{1\le i\le n,\ e\in S\cup\{0\}} |\hat{u}_{i,e} - \mu_{i,e}| \le \beta_q^{(S)}$$

with probability at least $1 - \delta/2$. Hence, on that event,

$$\max_{i,e} |\hat{u}_{i,e} - v_e(x_i)| \le \max_{i,e} |\hat{u}_{i,e} - \mu_{i,e}| + \max_{i,e} |\mu_{i,e} - v_e(x_i)| \le \beta_q^{(S)} + \eta_{\text{alias}},$$

which proves $\epsilon_{\text{desc}} \le \eta_{\text{alias}} + \beta_q^{(S)}$.

Substituting this bound into Theorem 3 gives

$$J_v(\rho_S, S) \ge F_v(S) - 2L\,\mathbb{E}\Big[\min_i d(X, x_i)\Big] - 2\epsilon_{\text{desc}} \ge F_v(S) - 2L\,\mathbb{E}\Big[\min_i d(X, x_i)\Big] - 2\eta_{\text{alias}} - 2\beta_q^{(S)}.$$

To guarantee routed regret at most τ relative to $F_v(S)$, it is enough that $2L\,\mathbb{E}[\min_i d(X, x_i)] + 2\eta_{\text{alias}} + 2\beta_q^{(S)} \le \tau$, i.e. that

$$\beta_q^{(S)} \le \frac{\tau - 2L\,\mathbb{E}[\min_i d(X, x_i)] - 2\eta_{\text{alias}}}{2}.$$

Squaring both sides and solving for q yields

$$\sigma^2 \frac{2\log(4(k+1)n/\delta)}{q} \le \frac{(\tau - 2L\,\mathbb{E}[\min_i d(X, x_i)] - 2\eta_{\text{alias}})^2}{4}, \qquad q \ge \frac{8\sigma^2 \log(4(k+1)n/\delta)}{(\tau - 2L\,\mathbb{E}[\min_i d(X, x_i)] - 2\eta_{\text{alias}})^2}.$$

This proves the rollout-complexity statement. □

Proposition 3 uses $k+1$ because its fixed library contains at most k skills plus the baseline. The full-matrix selection bound in Theorem 2 instead controls all $m+1$ columns.

An explicit bound on the expected nearest-neighbor radius also gives a sufficient number of calibration tasks.

Proposition 4 (Calibration-task complexity from descriptor coverage)**.** *Assume the conditions of Theorem 3. Define the expected nearest-neighbor radius by $r_n = \mathbb{E}[\min_{1\le i\le n} d(X, x_i)]$. Then $J_v(\rho_S, S) \ge F_v(S) - 2Lr_n - 2\epsilon_{\text{desc}}$. Assume, in addition, that the calibration descriptors satisfy the radius bound $r_n \le C\, n^{-1/d_{\text{eff}}}$ for some constants $C > 0$ and $d_{\text{eff}} > 0$. If a target routed regret level τ satisfies $\tau > 2\epsilon_{\text{desc}}$, then it is sufficient that*

$$n \ge \left(\frac{2LC}{\tau - 2\epsilon_{\text{desc}}}\right)^{d_{\text{eff}}},$$

and, if Proposition 3 also applies, it is sufficient that

$$n \ge \left(\frac{2LC}{\tau - 2\eta_{\text{alias}} - 2\beta_q^{(S)}}\right)^{d_{\text{eff}}}.$$

Proof. The first bound is just Theorem 3 with the shorthand $r_n = \mathbb{E}[\min_i d(X, x_i)]$. To ensure routed regret at most τ relative to $F_v(S)$, it is enough that $2Lr_n + 2\epsilon_{\text{desc}} \le \tau$. Using the assumed radius bound, $2Lr_n + 2\epsilon_{\text{desc}} \le 2LC\, n^{-1/d_{\text{eff}}} + 2\epsilon_{\text{desc}}$, so it is sufficient that $2LC\, n^{-1/d_{\text{eff}}} \le \tau - 2\epsilon_{\text{desc}}$. Because $\tau > 2\epsilon_{\text{desc}}$, this is equivalent to

$$n^{-1/d_{\text{eff}}} \le \frac{\tau - 2\epsilon_{\text{desc}}}{2LC},$$
$$n \ge \left(\frac{2LC}{\tau - 2\epsilon_{\text{desc}}}\right)^{d_{\text{eff}}}.$$

If Proposition 3 gives $\epsilon_{\text{desc}} \le \eta_{\text{alias}} + \beta_q^{(S)}$, then the same algebra with $2\epsilon_{\text{desc}} \le 2\eta_{\text{alias}} + 2\beta_q^{(S)}$ yields the second bound. □

Under the radius assumption in Proposition 4, more calibration tasks reduce r_n. More rollouts reduce the estimation term $\beta_q^{(S)}$. These are separate requirements for routing.

Remark 2. *Theorem 3 compares routed value with descriptor-conditional coverage. Task-level and descriptor-conditional coverage are*

$$F(S) = \mathbb{E}\left[\max_{e\in S\cup\{0\}} u_e(Z) - u_0(Z)\right] \qquad \textit{and} \qquad F_v(S) = \mathbb{E}\left[\max_{e\in S\cup\{0\}} v_e(X) - v_0(X)\right].$$

The next appendix bounds the difference between these quantities under a descriptor-sufficiency assumption.

E Descriptor sufficiency and limits of the model

To combine selection and routing guarantees, we first compare task-level coverage with the coverage available from a descriptor. We then give counterexamples for interacting skills, adaptive pool construction and unobserved calibration cells.

E.1 Information lost by the task descriptor

The two coverage objectives are

$$F(S) = \mathbb{E}\left[\max_{e\in S\cup\{0\}} u_e(Z) - u_0(Z)\right] \qquad \text{and} \qquad F_v(S) = \mathbb{E}\left[\max_{e\in S\cup\{0\}} v_e(X) - v_0(X)\right].$$

Proposition 5 (Descriptor-conditional oracle value is never larger than task-level oracle value)**.** *Assume the descriptor is a measurable function of the task, $X = x(Z)$. For every skill $e \in V \cup \{0\}$, define the descriptor-conditional utility $v_e(x) = \mathbb{E}[u_e(Z) \mid X = x]$, and for every library $S \subseteq V$ define*

$$F_v(S) = \mathbb{E}\left[\max_{e\in S\cup\{0\}} v_e(X) - v_0(X)\right].$$

Then $F_v(S) \le F(S)$.

Proof. Fix a library $S \subseteq V$. Conditional on X, we have

$$\begin{aligned}\max_{e\in S\cup\{0\}} v_e(X) - v_0(X) &= \max_{e\in S\cup\{0\}} \mathbb{E}[u_e(Z) \mid X] - \mathbb{E}[u_0(Z) \mid X] \\ &\le \mathbb{E}\left[\max_{e\in S\cup\{0\}} u_e(Z) \mid X\right] - \mathbb{E}[u_0(Z) \mid X] \\ &= \mathbb{E}\left[\max_{e\in S\cup\{0\}} u_e(Z) - u_0(Z) \mid X\right].\end{aligned}$$

The inequality uses the convexity of the pointwise maximum. Taking expectations yields

$$\begin{aligned} F_v(S) &= \mathbb{E}\left[\max_{e\in S\cup\{0\}} v_e(X) - v_0(X)\right] \\ &\le \mathbb{E}\left[\mathbb{E}\left[\max_{e\in S\cup\{0\}} u_e(Z) - u_0(Z) \mid X\right]\right] \\ &= \mathbb{E}\left[\max_{e\in S\cup\{0\}} u_e(Z) - u_0(Z)\right] \\ &= F(S). \end{aligned}$$

□

The inequality can be strict. Consider two equally probable tasks, $\Pr(Z = z_a) = \Pr(Z = z_b) = \frac{1}{2}$, with the same descriptor, $x(z_a) = x(z_b) = x_0$. Let the baseline be zero, $u_0(z_a) = u_0(z_b) = 0$, and let two skills specialize perfectly:

$$u_1(z_a) = 1, \qquad u_1(z_b) = 0, \qquad \text{and} \qquad u_2(z_a) = 0, \qquad u_2(z_b) = 1.$$

For the library $S = \{1, 2\}$, the task-level oracle value is $F(S) = 1$, while the descriptor-conditional values satisfy $v_1(x_0) = v_2(x_0) = \frac{1}{2}$, so $F_v(S) = \frac{1}{2}$.

Uniformly approximating each task utility by its descriptor-conditional mean bounds the difference between the objectives.

Proposition 6 (Uniform descriptor sufficiency implies a uniform oracle gap bound)**.** *Assume the descriptor is $X = x(Z)$, and that for some constant $\eta \ge 0$ every skill satisfies the pointwise approximation condition*

$$\sup_{e\in V\cup\{0\},\ z\in\mathcal{Z}} |u_e(z) - v_e(x(z))| \le \eta.$$

Then, for every library $S \subseteq V$,

$$|F(S) - F_v(S)| \le 2\eta, \qquad \textit{and consequently} \qquad 0 \le F(S) - F_v(S) \le 2\eta,$$

where the left inequality follows from Proposition 5. If, in addition, the baseline matches exactly, $u_0(z) = v_0(x(z))$ for all z, then the stronger bound $0 \le F(S) - F_v(S) \le \eta$ holds.

Proof. Fix a task z and write $x = x(z)$. For any library $S \subseteq V$,

$$\begin{aligned} &\left|\left(\max_{e\in S\cup\{0\}} u_e(z) - u_0(z)\right) - \left(\max_{e\in S\cup\{0\}} v_e(x) - v_0(x)\right)\right| \\ &\le \left|\max_{e\in S\cup\{0\}} u_e(z) - \max_{e\in S\cup\{0\}} v_e(x)\right| + |u_0(z) - v_0(x)| \\ &\le \sup_{e\in S\cup\{0\}} |u_e(z) - v_e(x)| + |u_0(z) - v_0(x)| \\ &\le \eta + \eta = 2\eta. \end{aligned}$$

Taking expectations over Z yields

$$\begin{aligned} |F(S) - F_v(S)| &= \left|\mathbb{E}\left[\max_{e\in S\cup\{0\}} u_e(Z) - u_0(Z)\right] - \mathbb{E}\left[\max_{e\in S\cup\{0\}} v_e(X) - v_0(X)\right]\right| \\ &\le \mathbb{E}\left[\left|\left(\max_{e\in S\cup\{0\}} u_e(Z) - u_0(Z)\right) - \left(\max_{e\in S\cup\{0\}} v_e(X) - v_0(X)\right)\right|\right] \\ &\le 2\eta. \end{aligned}$$

The inequality $F(S) - F_v(S) \ge 0$ is Proposition 5.

If $u_0(z) = v_0(x(z))$ for all z, the baseline term in the first display vanishes, so the per-task difference is bounded by η rather than 2η, and taking expectations gives $0 \le F(S) - F_v(S) \le \eta$. □

If $\eta = 0.05$, the proposition guarantees $|F(S) - F_v(S)| \le 0.10$ for every library S. This worst-case bound covers all size-k choices; it does not describe a particular routed policy.

The uniform sufficiency condition in Proposition 6 cannot be dropped. With an uninformative descriptor, the oracle gap can be made arbitrarily close to one.

Proposition 7 (Descriptor insufficiency can create an order-one oracle gap)**.** *For every integer $M \ge 2$, there exists a finite task space, a one-point descriptor space, and a library $S_M \subseteq V$ such that*

$$F(S_M) = 1, \qquad F_v(S_M) = \frac{1}{M}, \qquad \textit{and consequently} \qquad F(S_M) - F_v(S_M) = 1 - \frac{1}{M}.$$

Hence, for every $\epsilon > 0$, taking $M \ge 1/\epsilon$ gives an instance with $F(S_M) - F_v(S_M) \ge 1 - \epsilon$.

Proof. Let the task variable be $T \in \{1, \dots, M\}$ with distribution $\mathbb{P}(T = t) = \frac{1}{M}$. Let the descriptor space contain a single point, $\mathcal{X} = \{x_0\}$, and let the descriptor be constant: $X = x_0$ almost surely. Let the baseline satisfy $u_0(t) = 0$ for all t. Introduce M skills, indexed by $e \in \{1, \dots, M\}$, with utilities $u_e(t) = \mathbf{1}\{e = t\}$, and choose the library $S_M = \{1, \dots, M\}$. For every realized task type t, at least one selected skill matches it exactly, so $\max_{e \in S_M \cup \{0\}} u_e(t) = 1$. Therefore

$$F(S_M) = \mathbb{E}\left[\max_{e \in S_M \cup \{0\}} u_e(T) - u_0(T)\right] = \mathbb{E}[1 - 0] = 1.$$

Now compute the descriptor-conditional means. Since $X = x_0$ always, every skill $e \in S_M$ has $v_e(x_0) = \mathbb{E}[u_e(T) \mid X = x_0] = \mathbb{E}[u_e(T)] = \frac{1}{M}$, while $v_0(x_0) = 0$. Hence

$$F_v(S_M) = \mathbb{E}\left[\max_{e \in S_M \cup \{0\}} v_e(X) - v_0(X)\right] = \max_{e \in S_M \cup \{0\}} v_e(x_0) - 0 = \frac{1}{M}.$$

Subtracting the two displays gives $F(S_M) - F_v(S_M) = 1 - \frac{1}{M}$. Given $\epsilon > 0$, choose $M \ge \frac{1}{\epsilon}$; then $1 - \frac{1}{M} \ge 1 - \epsilon$. This proves the claim. □

For $M = 2$, $F(S_2) = 1$ while $F_v(S_2) = \frac{1}{2}$. The descriptor loses half of the oracle value because it cannot distinguish the two task types.

Combining the selection and routing theorems requires accounting for this loss of information. The following corollary separates calibration error, descriptor insufficiency and routing error.

Corollary 5 (Conditional end-to-end bridge for the selected library)**.** *Assume the event of Theorem 2, define $a_g = 1 - \frac{1}{e}$, and let the selected library be $\hat{S}_g$. Assume there exists a descriptor-sufficiency constant $\eta_{\mathrm{desc}} \ge 0$ such that*

$$\sup_{|S| \le k} (F(S) - F_v(S)) \le \eta_{\mathrm{desc}}.$$

Assume also the conditions of Theorem 3 hold for the fixed library $\hat{S}_g$, and define the expected nearest-neighbor radius $r_n = \mathbb{E}[\min_{1 \le i \le n} d(X, x_i)]$. Then

$$J_v(\rho_{\hat{S}_g}, \hat{S}_g) \ge a_g \max_{|S| \le k} F(S) - (1 + a_g)(\beta_n + 2\beta_q) - \eta_{\mathrm{desc}} - 2Lr_n - 2\epsilon_{\mathrm{desc}}.$$

If Proposition 6 holds with constant η, then the same conclusion holds with $\eta_{\mathrm{desc}} = 2\eta$, and with $\eta_{\mathrm{desc}} = \eta$ when the baseline matches exactly, $u_0(z) = v_0(x(z))$ for all z.

Proof. Combining Theorem 3, the descriptor-sufficiency assumption, and Theorem 2, in that order, yields

$$\begin{aligned} J_v(\rho_{\hat{S}_g}, \hat{S}_g) &\ge F_v(\hat{S}_g) - 2Lr_n - 2\epsilon_{\mathrm{desc}} \\ &\ge F(\hat{S}_g) - \eta_{\mathrm{desc}} - 2Lr_n - 2\epsilon_{\mathrm{desc}} \\ &\ge a_g \max_{|S| \le k} F(S) - (1 + a_g)(\beta_n + 2\beta_q) - \eta_{\mathrm{desc}} - 2Lr_n - 2\epsilon_{\mathrm{desc}}. \end{aligned}$$

If Proposition 6 holds, then $\sup_{|S| \le k} (F(S) - F_v(S)) \le 2\eta$, so one may substitute $\eta_{\mathrm{desc}} = 2\eta$. □

For example, if $a_g = 1 - \frac{1}{e} \approx 0.632$, $\beta_n + 2\beta_q = 0.05$, $\eta_{\mathrm{desc}} = 0.08$, and $2Lr_n + 2\epsilon_{\mathrm{desc}} = 0.04$, then the corollary gives $J_v(\rho_{\hat{S}_g}, \hat{S}_g) \ge 0.632 \max_{|S| \le k} F(S) - 0.17$.

E.2 COUNTEREXAMPLES WHEN ASSUMPTIONS ARE REMOVED

If a task requires two skills together, its value need not be submodular. The next proposition shows why we assume atomic, one-skill-per-task invocation.

Proposition 8 (AND-gate interaction breaks submodularity and the pointwise-max model)**.** *There exists a normalized monotone set function $G : 2^V \to [0, 1]$ on $V = \{A, B\}$ such that*

$$G(\emptyset) = 0, \qquad G(\{A\}) = 0, \qquad G(\{B\}) = 0, \qquad G(\{A, B\}) = 1,$$

and G is not submodular. Moreover, G cannot be represented in the atomic pointwise-maximum form

$$G(S) = \max_{e \in S \cup \{0\}} u_e - u_0$$

for any fixed utilities $u_A,\ u_B,\ u_0 \in \mathbb{R}$.

Proof. The marginal gain of adding B to the empty set is $G(\{B\}) - G(\emptyset) = 0$, while the marginal gain of adding B to $\{A\}$ is $G(\{A, B\}) - G(\{A\}) = 1$. Therefore $G(\{B\}) - G(\emptyset) < G(\{A, B\}) - G(\{A\})$, which violates diminishing returns. Hence G is not submodular.

For the representation claim, suppose by contradiction that $G(S) = \max_{e \in S \cup \{0\}} u_e - u_0$. Since $G(\{A\}) = 0$, we must have $\max(u_A, u_0) - u_0 = 0$, which implies $u_A \le u_0$. Likewise, from $G(\{B\}) = 0$, we obtain $u_B \le u_0$. Then

$$G(\{A, B\}) = \max(u_A, u_B, u_0) - u_0 = u_0 - u_0 = 0,$$

contradicting $G(\{A, B\}) = 1$. □

If skill A opens a file and skill B parses it, neither alone solves the task, but the ordered pair does. This AND-gate dependency falls outside the SkillCover model.

Remark 3 (Greedy can be arbitrarily bad under interaction)**.** *Once submodularity fails, exact greedy loses its approximation guarantee entirely, not just up to constants. Extend the AND-gate construction by a decoy: for every $\varepsilon \in (0, 1)$, let $V = \{a, b, c\}$ with*

$$G(\emptyset) = G(\{a\}) = G(\{b\}) = 0, \qquad G(\{c\}) = G(\{a, c\}) = G(\{b, c\}) = \varepsilon,$$
$$G(\{a, b\}) = G(\{a, b, c\}) = 1.$$

This G is monotone but not submodular. Under budget $k = 2$, greedy first picks the decoy c because $G(\{c\}) = \varepsilon > 0 = G(\{a\}) = G(\{b\})$, and the second pick adds nothing since $G(\{a, c\}) = G(\{b, c\}) = \varepsilon$. Greedy therefore attains value ε while the optimal library $\{a, b\}$ attains 1, so the greedy-to-optimal ratio is ε, which can be made arbitrarily small. This is the formal counterpart of the interaction stress test in Appendix I.6: outside the pointwise-max regime, no multiplicative guarantee for greedy survives.

Allowing the pool to depend on calibration outcomes can also invalidate the selection guarantee.

Proposition 9 (Adaptive pool generation can interpolate the calibration sample)**.** *For every calibration sample size $n \ge 1$, there exists a data-dependent pool generator that, after observing calibration tasks $D_n = \{Z_1, \ldots, Z_n\}$, produces a candidate pool $V(D_n) = \{e_{D_n}\}$ with the following property:*

$$F_n(\{e_{D_n}\}) = 1 \qquad \textit{and} \qquad F(\{e_{D_n}\}) = 0 \qquad \textit{almost surely}.$$

Hence $F_n(\{e_{D_n}\}) - F(\{e_{D_n}\}) = 1$ almost surely.

For a continuous-task example, let $Z \sim \mathrm{Unif}[0, 1]$ and set the baseline to zero, $u_0(z) = 0$. After observing D_n, define a memorizer, $u_{e_{D_n}}(z) = \mathbf{1}\{z \in D_n\}$. It scores 1 on every calibration task and 0 on a fresh draw with probability one.

Proof. By construction, $u_{e_{D_n}}(Z_i) = 1$ for every $i \in \{1, \ldots, n\}$. Therefore

$$F_n(\{e_{D_n}\}) = \frac{1}{n} \sum_{i=1}^{n} \big[\max\big(u_{e_{D_n}}(Z_i), u_0(Z_i)\big) - u_0(Z_i)\big] = \frac{1}{n} \sum_{i=1}^{n} 1 = 1.$$

For the population value, a fresh draw $Z \sim \mathrm{Unif}[0,1]$ satisfies $\Pr(Z \in D_n) = 0$ almost surely because D_n is finite. Hence $u_{e_{D_n}}(Z) = 0$ almost surely, and thus

$$F(\{e_{D_n}\}) = \mathbb{E}\big[\max\big(u_{e_{D_n}}(Z), u_0(Z)\big) - u_0(Z)\big] = \mathbb{E}[0] = 0.$$

Therefore $F_n(\{e_{D_n}\}) - F(\{e_{D_n}\}) = 1$ almost surely. □

Here the candidate class is chosen from the observed sample, so the finite-family concentration argument in Theorem 2 no longer applies.

Finally, unobserved cells can prevent a selector from identifying the optimal library.

Proposition 10 (Partial-matrix indistinguishability)**.** *There exist two worlds $W^{(a)}$ and $W^{(b)}$ with the same queried cells but different optimal libraries. In particular, let $V = \{1, 2\}$, let the budget be $k = 1$, let the baseline satisfy $u_0(z_1) = u_0(z_2) = 0$, and let the task distribution be $\Pr(Z = z_1) = \Pr(Z = z_2) = \frac{1}{2}$. Suppose the queried set is $Q = \{(z_1, 1), (z_1, 2)\}$, and assume both worlds agree on the queried cells:*

$$u_1^{(a)}(z_1) = u_1^{(b)}(z_1) = 1, \qquad u_2^{(a)}(z_1) = u_2^{(b)}(z_1) = 1.$$

Now define the unqueried cells in world $W^{(a)}$ by $u_1^{(a)}(z_2) = 1$ and $u_2^{(a)}(z_2) = 0$, and in world $W^{(b)}$ by $u_1^{(b)}(z_2) = 0$ and $u_2^{(b)}(z_2) = 1$. Then

$$\arg\max_{|S|\le 1} F^{(a)}(S) = \{\{1\}\}, \qquad \textit{while} \qquad \arg\max_{|S|\le 1} F^{(b)}(S) = \{\{2\}\}.$$

Therefore no selector that only observes Q can identify the correct optimal library in both worlds. Moreover, for any randomized selector that uses only the revealed data, the average regret under the uniform prior $\mathbb{P}(W^{(a)}) = \mathbb{P}(W^{(b)}) = \frac{1}{2}$ is at least $\frac{1}{4}$.

The two worlds share the observed first row, $(z_1):\ (u_1, u_2) = (1, 1)$, and differ only on the hidden second row:

$$(z_2) \text{ in } W^{(a)}:\ (u_1, u_2) = (1, 0), \qquad (z_2) \text{ in } W^{(b)}:\ (u_1, u_2) = (0, 1).$$

The optimal singleton flips, even though the queried evidence is identical.

Proof. In world $W^{(a)}$,

$$F^{(a)}(\{1\}) = \frac{1}{2}\cdot 1 + \frac{1}{2}\cdot 1 = 1, \qquad \text{while} \qquad F^{(a)}(\{2\}) = \frac{1}{2}\cdot 1 + \frac{1}{2}\cdot 0 = \frac{1}{2},$$

so $\arg\max_{|S|\le 1} F^{(a)}(S) = \{\{1\}\}$. In world $W^{(b)}$,

$$F^{(b)}(\{1\}) = \frac{1}{2}\cdot 1 + \frac{1}{2}\cdot 0 = \frac{1}{2}, \qquad \text{while} \qquad F^{(b)}(\{2\}) = \frac{1}{2}\cdot 1 + \frac{1}{2}\cdot 1 = 1,$$

so $\arg\max_{|S|\le 1} F^{(b)}(S) = \{\{2\}\}$.

Because the queried cells are identical in the two worlds, any procedure that only receives Q observes exactly the same data in $W^{(a)}$ and $W^{(b)}$. Therefore it must return the same output in both worlds. Since the optimal output differs, no such procedure can be uniformly correct.

For the randomized claim, the output distribution of any selector that sees only Q is likewise identical in both worlds. Write $p_1 = \mathbb{P}(\hat{S} = \{1\})$, $p_2 = \mathbb{P}(\hat{S} = \{2\})$, and $p_0 = \mathbb{P}(\hat{S} = \emptyset)$, with $p_0 + p_1 + p_2 = 1$. From the values computed above, the regret of $\{2\}$ in $W^{(a)}$ and of $\{1\}$ in $W^{(b)}$ is $\frac{1}{2}$, and the regret of $\emptyset$ is 1 in both worlds. Hence the average regret under the uniform prior is

$$\frac{1}{2}\left(\frac{p_2}{2} + p_0\right) + \frac{1}{2}\left(\frac{p_1}{2} + p_0\right) = \frac{1}{4}(p_1 + p_2) + p_0 = \frac{1}{4} + \frac{3}{4}\,p_0 \ge \frac{1}{4}.$$

□

Propositions 8–10 identify the roles of atomic utilities, a fixed pool and a full matrix. The single-skill selection studies use the corresponding coverage objective; mounted execution can involve interactions and is evaluated separately. The stress tests in Appendix I.6 examine departures from these assumptions.

Table 4: Proofs and synthetic diagnostics for the formal claims. Proofs appear in this and the preceding appendices; the diagnostics are reported in Section 4.

Claim	Proof location	Empirical target in Section 4	Status
Lemma 1	Appendix B; bounded normalization after clipping	Implementation audit and matrix logging checks; appendix-only diagnostic	Proved
Lemma 2	Appendix B; diminishing returns of the pointwise maximum	Redundant-clone world and marginal-gain diagnostics	Proved
Lemma 4	Appendix C; uniform convergence over N_k libraries	n-sweep consistency study and held-out coverage gaps	Proved
Lemma 5	Appendix C; stability under entrywise perturbations	Controlled perturbation diagnostic	Proved
Lemma 6	Appendix C; transfer under uniform objective error	Greedy-transfer diagnostic	Proved
Theorem 1	Appendix B; greedy approximation and singleton separation	Redundant-clone singleton/OPT panel and latent held-out sweeps	Proved; synthetically supported
Theorem 2	Appendix C; Lemmas 4–6	Trimmed n- and q-scaling fits; qualitative because one path uses Gaussian cell noise	Proved; qualitatively supported
Theorem 3	Appendix D; Lipschitz descriptor utilities	Clean Lipschitz world and descriptor-corruption panel	Proved; conditionally supported
Proposition 5	This section; Jensen-type ordering $F_v(S) \leq F(S)$	No separate figure; used to interpret the oracle/deployment decomposition	Proved
Proposition 6	This section; uniform descriptor sufficiency implies $\lvert F(S) - F_v(S)\rvert \leq 2\eta$	No direct experiment; assumption-level bridge only	Proved
Corollary 5	This section; conditional composition of Theorems 2 and 3	No end-to-end empirical claim; included only as a conditional bridge	Proved
Propositions 8–10	This section; interaction, adaptive-pool, and partial-matrix counterexamples	Out-of-regime appendix stress tests and discussion limitations	Proved as counterexamples

E.3 Proofs and associated diagnostics

Table 4 maps formal claims to their proofs and associated synthetic diagnostics. The real SkillsBench and MCPAgentBench studies are reported in Sections 4.1 and 4.4.

The population coverage objective remains

$$F(S) = \mathbb{E}\left[\max_{e \in S \cup \{0\}} u_e(Z) - u_0(Z)\right].$$

The noisy-greedy theorem controls $F(\hat{S}_g)$; the routing theorem controls $J_v(\rho_S, S)$ relative to $F_v(S)$. Corollary 5 combines them only after bounding the descriptor-sufficiency gap $F(S) - F_v(S)$.

The remaining appendices provide worked examples, study protocols, implementation details and numerical results.

F WORKED EXAMPLES AND EVALUATION GUIDE

The examples below illustrate calibration, greedy selection and the held-out coverage metrics. Table 5 then relates the evaluation studies to the analysis.

F.1 WORKED EXAMPLES FOR THE METHOD

Cell estimation. If $q = 3$ and one task–skill cell yields rollout utilities 0.8, 0.4, and 0.7, the stored cell estimate is $\hat{u}_{i,e} = \frac{1}{3}(0.8 + 0.4 + 0.7) = 0.633$. The selector needs only this mean.

Marginal coverage (Lemma 2). Consider one task z with baseline utility $u_0(z) = 0.2$ and two candidate skills with utilities $u_1(z) = 0.7$ and $u_2(z) = 0.6$. With only skill 1, the task value is $f_z(\{1\}) = 0.7 - 0.2 = 0.5$; with both skills, it is still $f_z(\{1, 2\}) = 0.5$. The second skill adds no value because the first has higher utility.

Greedy selection (Algorithm 1). Consider $n = 2$ calibration tasks and $k = 2$. Let the baseline utilities be $\hat{u}_{1,0} = \hat{u}_{2,0} = 0.2$, and let three skills have calibration values $(\hat{u}_{1,1}, \hat{u}_{2,1}) = (0.7, 0.2)$, $(\hat{u}_{1,2}, \hat{u}_{2,2}) = (0.6, 0.4)$, and $(\hat{u}_{1,3}, \hat{u}_{2,3}) = (0.3, 0.9)$. The singleton empirical coverage values are

$$\hat{F}_n(\{1\}) = \tfrac{1}{2}\,[0.5 + 0] = 0.25, \qquad \hat{F}_n(\{2\}) = 0.30, \qquad \hat{F}_n(\{3\}) = 0.40,$$

so the first greedy choice is $e_0 = 3$. The marginal gain of adding skill 1 to $\{3\}$ is $\hat{F}_n(\{3, 1\}) - \hat{F}_n(\{3\}) = \frac{1}{2}[(0.7 - 0.3) + 0] = 0.20$, while for skill 2 it is $\frac{1}{2}[(0.6 - 0.3) + 0] = 0.15$, so the second greedy choice is $e_1 = 1$.

The second step favors skill 1, whose gain on the first task is larger after skill 3 has been retained.

F.2 NUMERIC INSTANTIATION OF THEOREM 2

Take $m = 10$, $k = 4$, $n = 512$, $q = 64$, $\delta = 0.1$, and conditionally independent clipped rollouts with a common mean, so that $\sigma = \frac{1}{2}$. Then $N_k = \sum_{j=0}^{4} \binom{10}{j} = 386$ and $a_g \approx 0.632$. The two error terms are approximately

$$\beta_n \approx \sqrt{\frac{\log(4 \cdot 386/0.1)}{2 \cdot 512}} \approx 0.097, \qquad \beta_q \approx \frac{1}{2}\sqrt{\frac{2\log(4 \cdot 11 \cdot 512/0.1)}{64}} \approx 0.310.$$

The excess term is $(1 + a_g)(\beta_n + 2\beta_q) \approx 1.632 \times 0.718 \approx 1.17$, larger than the full range of the coverage objective. The resulting lower bound is therefore uninformative at these parameters. Section 4 examines scaling trends and held-out coverage gaps rather than using the bound as a numerical performance prediction.

F.3 WORKED EXAMPLE FOR THE HELD-OUT METRICS

The archive abbreviates SkillCover as SC and sometimes calls the coverage gap "regret." We use "routing regret" for the loss incurred by dispatching within a fixed library. The reference for each coverage gap is stated with the experiment.

If $F_{\text{holdout}}(S^{\star}_{\text{holdout}}) = 0.82$ and $F_{\text{holdout}}(S_{\text{SC}}) = 0.81$, then $\text{Gap}(S_{\text{SC}}) = 0.01$. If instead $F_{\text{holdout}}(S_{\text{singleton}}) = 0.64$, then $\text{Adv}(S_{\text{SC}}, S_{\text{singleton}}) = (0.82 - 0.64) - (0.82 - 0.81) = 0.17$.

F.4 HOW THE STUDIES RELATE TO THE ANALYSIS

Table 5 records which experiment in Section 4 targets which formal claim.

Table 5: The analysis and its associated evaluations. Synthetic studies examine the stated mechanisms and assumptions. Real-matrix and mounted-execution studies provide empirical comparisons with the scope limits described in Section 4. Supporting lemma diagnostics appear in Appendix I.

Claim	Empirical target	Reported quantity	Status
Theorem 1	Redundant-clone world	singleton/OPT ratio, greedy/OPT	exact synthetic support
Theorem 1	Latent held-out sweeps	$\mathrm{Gap}(S)$, Adv	synthetic support
Theorem 2	n- and q-sweeps	slope and rank-correlation vs theory terms	qualitative synthetic support
Theorem 3	Lipschitz descriptor worlds	routing regret under clean/corrupted descriptors	conditional synthetic support
Lemma 4, Lemma 5, Lemma 6	perturbation diagnostics	max-stability and transfer checks	Appendix I
Theorems 1,2,3	MCPAgentBench (real)	held-out coverage gaps and routing	fixed-task diagnostics
Theorems 1,2	SkillsBench (real, self-run)	observed-matrix coverage and mounted-library performance	fixed-task diagnostics

G SkillsBench protocols and supplementary results

This appendix documents the candidate pool, archived observations and supplementary results for the SkillsBench study in Section 4.1. The tables regenerate deterministically from the compact observation bundle via `experiments/skillsbench_reporting_export.py`. The export reproduces historical controls on the archived pre-recovery bundle, verifies that measured entries are preserved, and checks unaffected model means. Newly measured Sonnet results may differ from the older analysis.

G.1 Candidate pool and archived observations

Pool construction and exclusions. From the SkillsBench release, the original directory-reuse filter found 16 candidate directories present in at least two tasks. A subsequent entry-point audit found that `licenses` contains only `obspy.LICENSE` and `seisbench.LICENSE`, with no `SKILL.md`. The revised selection analysis excludes that directory and uses 15 valid skills: xlsx, pdf, pptx, docx, browser-testing, react-best-practices, threejs, obj-exporter, pddl-skills, power-flow-data, economic-dispatch, dc-power-flow, obspy-data-api, seismic-picker-selection, and seisbench-model-api.

Two seismology tasks (earthquake phase association and seismic phase picking) require an external model server that was unreachable throughout the study and are excluded, leaving 30 tasks. Their three valid seismology skills remain in the pool, although the tasks that supplied them are absent. Where tasks ship content variants of a same-named skill, the variant shared by the most tasks is canonical.

The fourth base model was originally Llama-4-Maverick, whose tool calls under this scaffold emit plain text and never execute. It was replaced by DeepSeek-V3.2 to keep four model families. The harness strips each task's bundled skills before mounting the assigned condition, and a fidelity guard rejects mismatched mounted skill catalogs.

Completion of missing observations. The original matrix contained 4,065 outcomes, with 15 missing Sonnet entries across nine tasks; two mounted-library outcomes were also missing. A supplemental campaign on 17 September 2026 recovered all 17 outcomes: eight successes and nine scored failures. The first valid verifier score completed each missing cell, including agent timeouts that the verifier scored zero. Unscored setup or verifier failures could be retried; previously observed rewards were never replaced. The complete archive contains 4,080 matrix and 480 mounted-library outcomes. Excluding the 240 license-directory observations leaves 3,840 selection observations across 15 valid skills plus no-skill, all 30 tasks and both repetitions for every model. No reward is

Table 6: Greedy-minus-singleton point estimates under the historical tie rules and common candidate priority, averaged over both rollout partitions on all 30 tasks per model. The control changes only singleton tie handling.

	$k = 2$ advantage		$k = 4$ advantage	
Model	Historical	Common priority	Historical	Common priority
Claude Sonnet 5	$+0.017$	$+0.000$	-0.033	$+0.000$
GLM-5	$+0.117$	$+0.117$	$+0.083$	$+0.083$
Qwen3-Coder	$+0.000$	-0.033	$+0.033$	-0.017
DeepSeek-V3.2	$+0.000$	-0.017	-0.050	-0.017

imputed or overwritten. The original and intermediate bundles and the legacy 16-directory reanalysis are preserved.

Recovery matched the archived task-source hashes, model identifier, core agent commit, and SDK/tool versions. Transitive dependencies differ from the original environment. An isolated repair changes process cleanup and verifier-setup deadlines, without changing task grading logic or agent inputs and time limits. These differences and each recovered result's hash and timestamp are recorded in the bundle metadata. Jobs were ordered by their recorded task time limits, then task, condition, and repetition.

G.2 Matrix sensitivity analyses

Tie-rule sensitivity. The primary analysis retains the historical selection implementation on the corrected 15-skill pool. Table 6 gives singleton the same candidate priority as greedy when calibration scores tie. The control changes only singleton tie handling; rollouts and greedy selections are unchanged. GLM-5's point advantages are unchanged, while Sonnet's become zero at both $k = 2$ and $k = 4$. Qwen3-Coder's $k = 4$ comparison changes sign. Some small observed differences therefore depend on tie handling; the controls do not establish a population advantage.

Exploratory task-disjoint refitting. We exclude one task at a time, average both repetitions on each of the other 29 tasks, and fit size-four libraries over the 15 valid skills using common candidate priority. The excluded task is evaluated using its two-repetition mean matrix. Greedy-minus-singleton coverage is -0.017 for DeepSeek-V3.2, 0 for Qwen3-Coder, $+0.050$ for Sonnet, and -0.067 for GLM-5. These results are exploratory because the task outcomes had already been examined. The folds share most training data and are not treated as independent experiments. GLM's positive same-task rollout comparison does not persist under this protocol.

Results by task domain. Table 7 groups the $k = 4$ primary results by task domain, pooling all 120 (model, task) pairs under each model's own calibrated selection. Each pair has equal weight. These summaries are descriptive, with especially small samples in the less frequent domains.

Table 7: Held-out matrix coverage relative to no-skill at $k = 4$, grouped by task domain. Counts in parentheses are (model, task) pairs. Each model uses its own calibrated selection; bold marks the best selector per column.

Method	Office (44)	Finance (28)	SWE (16)	Indust. (12)	Math (8)	Media (8)	Sci. (4)	**ALL** (120)
Greedy	**0.250**	0.179	0.156	0.125	0.062	**0.312**	0.625	**0.212**
Singleton	0.216	0.196	0.125	**0.167**	0.062	0.250	0.750	0.204
Diversity	0.170	**0.214**	0.125	0.083	**0.188**	0.188	**0.875**	0.192
k-medoids	0.193	0.143	0.156	**0.167**	**0.188**	0.188	0.375	0.179
Random	0.226	0.152	**0.164**	0.137	0.164	0.241	0.484	0.197

Skill-level diagnostics. The canonical export restricts descriptive gain counts to the 15 valid single-skill conditions and the primary task set, excluding the license directory and mounted-library arms. Positive or negative gain relative to no-skill describes a task–skill observation; it is not a direct

measure of redundancy between skills. We therefore do not infer a model's redundancy ranking or a causal failure mechanism from the fraction of negative gains.

G.3 Historical mounted-library comparison

The legacy selected arm uses the directories fixed for the original deployment campaign: docx, browser-testing, economic-dispatch, pptx (DeepSeek-V3.2); docx, pptx, licenses, economic-dispatch (Qwen3-Coder); pdf, obj-exporter, power-flow-data, xlsx (Sonnet); and pptx, threejs, obj-exporter, xlsx (GLM-5). Qwen's bundle therefore contains three valid skills and one license-only directory. We evaluate these recorded configurations without reselecting from the completed matrix (Table 8).

The paired intervals include zero in every model. The best observed single-skill column uses hindsight across 15 valid skill conditions and should not be used to attribute the performance difference solely to routing. Static metadata counts for prompt injection also differ from token use in this filesystem-discovery scaffold.

Table 8: Historical SkillsBench deployment success over 30 tasks and two repetitions per arm. Legacy full contains 15 skills plus a license directory; legacy selected contains four skills, except Qwen's three skills plus that directory. Best observed (15) is the post-hoc per-task/repetition best single-skill outcome. The final column reports the paired task-bootstrap 90% interval.

Model	No skills	Legacy full	Legacy selected	Best observed (15)	Selected − full (90% CI)
Claude Sonnet 5	0.417	0.533	0.533	0.767	+0.000 [−0.100, +0.100]
GLM-5	0.117	0.333	0.267	0.467	−0.067 [−0.150, +0.017]
Qwen3-Coder	0.017	0.083	0.083	0.300	+0.000 [−0.067, +0.050]
DeepSeek-V3.2	0.050	0.117	0.150	0.383	+0.033 [−0.050, +0.117]

G.4 Mounted-library transfer to new tasks

We select each library from the complete calibration matrix before observing any new-task outcomes. For each model, we average both repetitions within each calibration task and skill, retain the measured no-skill baseline, and select four of the 15 valid skills. Greedy and Singleton use the same sorted candidate priority when scores tie. Table 10 records the selected sets. GLM's sets coincide, so its two selector labels use the same 28 executions; their zero contrast is structural, not an independently measured equivalence result.

Task sampling and execution. We retain the SkillsBench source revision used for calibration and exclude all 32 tasks touched by the original reusable-skill filter, including the two seismology tasks omitted from that analysis. Within each of its seven categories, eligible task names are ranked by SHA256 of a fixed prefix followed by the name, and the first two are selected (Table 9). The manifest records the prefix, complete candidate ranking and exclusions. Before any new outcomes, we exclude `travel-planning` because its instructions require native skills that the experimental scaffold removes. Category balance and the eligibility filter limit this transfer assessment: the tasks are not a representative random sample of deployment tasks.

Task/model/repetition blocks and arm order within blocks are randomized once and frozen. Each model receives two repetitions of each new task under Base, all 15 candidate skills and the two selected libraries. Identical libraries share observations, giving $14 \times 2 \times (4+4+4+3) = 420$ unique executions. Native task skills are removed; candidate skills are mounted together and advertised in the agent's skill catalog. OpenHands background skills remain available in every arm, so Base denotes no added candidate-pool skills and $k = 4$ counts only retained candidates. The model identifiers, task files, staged libraries, source revision and execution budgets are recorded in the manifest. No new-task outcome is used to refit a library, replace a task or select an attempt.

Scores and uncertainty. The first finite native verifier score in $[0, 1]$ without a verifier error completes an experimental cell. Zero, partial credit and graded agent timeouts are terminal. Full-credit success equals one exactly when the native score is 1. The `tictoc-unnecessary-abort-detection` verifier awards partial credit; the other thirteen

Table 9: The 14 transfer tasks, fixed before their outcomes were observed. Each original category contributes two tasks outside the calibration task filter.

Category	New tasks
Office	`edit-pdf`
	`enterprise-information-search`
Finance	`econ-detrending-correlation`
	`sec-financial-report`
SWE	`fix-build-google-auto`
	`tictoc-unnecessary-abort-detection`
Indust.	`dynamic-object-aware-egomotion`
	`r2r-mpc-control`
Math	`bike-rebalance`
	`paratransit-routing`
Media	`multilingual-video-dubbing`
	`video-silence-remover`
Sci.	`exoplanet-detection-period`
	`gravitational-wave-detection`

Table 10: Four-candidate libraries fitted on all 30 calibration tasks using both repetitions and common candidate priority. Listed order is alphabetical; mounted contents are fixed. The common OpenHands background catalog is outside this candidate budget.

Model	Selector	Four retained skills
Claude Sonnet 5	Greedy	obj-exporter, pdf, react-best-practices, xlsx
Claude Sonnet 5	Singleton	pdf, pptx, react-best-practices, seismic-picker-selection
GLM-5	Greedy	obj-exporter, pptx, threejs, xlsx
GLM-5	Singleton	obj-exporter, pptx, threejs, xlsx
Qwen3-Coder	Greedy	docx, economic-dispatch, pptx, seismic-picker-selection
Qwen3-Coder	Singleton	docx, pddl-skills, pdf, react-best-practices
DeepSeek-V3.2	Greedy	browser-testing, docx, economic-dispatch, pptx
DeepSeek-V3.2	Singleton	browser-testing, docx, economic-dispatch, threejs

wrappers return binary outcomes. Our initial result reader incorrectly rejected fractional scores and triggered redundant retries. We corrected it after discovering these outcomes, retaining the earliest native score for every cell and archiving later attempts only for provenance and costs. This clarification followed the discovery of fractional outcomes; binary native scores were not a preregistered assumption.

For each model and contrast, we pair outcomes by task and repetition, average the two differences within each task, and report the mean over 14 tasks. Percentile 95% intervals use 10,000 bootstrap resamples of whole tasks with the libraries held fixed. They are descriptive, conditional intervals without correction for multiple comparisons or library re-selection uncertainty. All 420 planned cells have valid scores; no missing outcome is imputed. Table 11 reports mean native reward under the same pairing and resampling rule.

Results and scope. Sonnet's full-credit Greedy-minus-Singleton difference is 0.107, with interval $[-0.036, 0.250]$; its native-score difference is 0.136, with interval $[-0.007, 0.286]$. DeepSeek's native-score difference is -0.059, Qwen's 0.036, and GLM's structural difference is zero. The comparison remains unresolved when partial credit is retained. Full-credit Greedy-minus-all-15 differences are 0.000, 0.036, -0.036, and -0.143 for DeepSeek, Qwen, Sonnet and GLM, respectively; each paired 95% interval includes zero. Sonnet succeeds in 13/28 executions with Greedy, versus 14/28 with either all candidates or Base. GLM succeeds in 4/28 with its selected set and 8/28 with all candidates. Fewer retained candidates therefore do not establish lower execution costs, performance equivalence or an improvement over Base. Mounted success reflects catalog exposure and possible skill interactions; the selection theorem evaluates a single-skill maximum.

Table 11: Native-score sensitivity for the transfer study. Entries are mean verifier reward in $[0, 1]$, including partial credit. Base and the candidate arms share the background catalog. Full-credit success counts appear in Table 2. † GLM's identical selector sets share observations.

Model	Base	All 15	Greedy 4	Singleton 4	$\Delta_{\mathrm{G-S}}$	95% CI
Claude Sonnet 5	0.500	0.546	0.495	0.359	$+0.136$	$[-0.007, +0.286]$
GLM-5	0.163	0.291	0.179	0.179	$0^{\dagger}$	shared†
Qwen3-Coder	0.041	0.041	0.075	0.039	$+0.036$	$[-0.071, +0.143]$
DeepSeek-V3.2	0.045	0.077	0.089	0.148	-0.059	$[-0.214, +0.037]$

Post-hoc catalog and skill use audit. After completing the study, we inspect the archived initial requests and tool traces for all 420 retained executions. Every initial catalog contains exactly the configured candidate set plus the same 64 background skill entries; the background entry metadata hashes also agree. These background entries include skills outside our 15-candidate pool and are not covered by its retention budget. The archived `no-skill` condition is therefore labelled Base in the transfer tables. This audit changed neither the frozen libraries nor the scores.

Across all executions, seven explicit skill invocations are recorded: five calls to `pdf` in the all-candidate arm and two calls to the background `agent-memory` skill in Base. The 196 unique Greedy/Singleton executions record none, and their tool input/display metadata contains no explicit reference to the mounted skill path or a `SKILL.md` file. The invocation counter excludes ordinary filesystem reads, and the path scan is only a limited proxy for such reads. Catalog descriptions are visible without invoking a skill. The traces do not show that selected skill bodies were invoked to produce the gains, but cannot rule out influence from skill information.

Discordant task outcomes. We include every nonshared task/repetition pair in the descriptive comparison: 84 pairs, of which 14 differ in full-credit success and another two differ only in native partial credit. Greedy wins, Singleton wins and full-credit ties number $1/3/24$ for DeepSeek, $2/1/25$ for Qwen and $5/2/21$ for Sonnet. GLM's 28 shared pairs are excluded from these win/loss counts. The source package records all 16 discordant pairs, their verifier checks and trace hashes; none of their 32 executions has a recorded terminal agent error. Partial credit remains distinct from a binary failure.

The checks identify varied output errors. DeepSeek's economic-correlation task reverses its winning selector between repetitions; each losing output fails the numerical correlation check. On Sonnet's PDF task, Singleton fails the signature check, although both libraries advertise `pdf` and neither invokes it explicitly. Sonnet's TicToc scores are $1.00/0.05$ for Greedy/Singleton in one repetition and $0.85/1.00$ in the other; the task's policy-ladder score is separate from its three passing structural tests. These cases identify failed output requirements without attributing them to skill redundancy, complementarity or routing. Repetition labels match experimental blocks, not provider random seeds.

Runtime recovery and accounting. The archive preserves successive infrastructure repairs to dependency pins, per-rollout Docker image tags, isolated container cleanup, setup permissions, initialization waiting, pre-verifier log publication and provider authentication. These repairs leave task inputs, verifier logic, selected libraries and agent execution budgets fixed. Only unscored attempts may be retried, under the documented uniform ceilings of three, then six, then nine attempts per cell. Credential failures subsequently pause launches before allocating another attempt. Every retained score comes from the first valid native result, and all 142 scores present before the authentication recovery remain unchanged.

The new-task campaign, its retired pre-correction pilot and separately logged connectivity checks total approximately $1,077.08 in recorded or estimated model-call usage. Including the earlier Sonnet recovery gives $1,238.28. These totals include failed and redundant attempts with available usage; attempts lacking telemetry remain unknown rather than being assigned zero. They exclude the original historical campaigns and host-compute charges and are not a reconciliation to a provider invoice. The source package records phase totals and provenance separately. No outcomes from the retired license-directory pilot contribute to this transfer study.

Table 12: Fixed tasks for the fresh-calibration diagnostic. Columns list disjoint sets; rows do not imply task pairing. All identities were observed before this diagnostic.

Calibration tasks	Evaluation tasks
offer-letter-generator	invoice-fraud-detection
court-form-filling	jpg-ocr-stat
pptx-reference-formatting	organize-messy-files
weighted-gdp-calc	paper-anonymizer
financial-modeling-qa	sales-pivot-analysis
exceltable-in-ppt	xlsx-recover-data

G.5 Fresh-calibration document-task diagnostic

Question and task scope. Following the mixed deployment results, we examine whether additional calibration within a narrower document-task scope changes the Greedy–Singleton comparison. The two models are DeepSeek-V3.2 and Sonnet, with the same 15 valid candidate packages and a retention budget of four. Table 12 lists six calibration tasks and six evaluation tasks. All twelve identities appeared in earlier experiments. Evaluation tasks are excluded from this study's fitting, but they are not an untouched test set. The design is an exploratory follow-up to earlier noise and task-shift analyses.

The six fixed calibration tasks in Table 12 come from an earlier diagnostic of skill loading. Their historical scores are not reused. Evaluation uses the remaining original-30 office/finance tasks with at least one native skill overlapping the pool, a native agent limit of at most 1,800 seconds, and no explicit URL, download or latest-information request in the instruction. The frozen manifest records the exact source-based rule and task hashes.

Some tasks also ship auxiliary skills outside the candidate pool. We disable task-native skills and expose the assigned candidates through ordinary catalog discovery, alongside 64 OpenHands background entries. The background is outside the four-candidate retention budget.

Acquisition and selection. We collect four new repetitions for every model, calibration task and single-candidate/Base condition: $2 \times 6 \times 16 \times 4 = 768$ executions. Historical rewards are not reused. Libraries are fitted only after every cell has a native score and passes configuration checks. Both selectors receive the same task–skill means and baseline means, with common alphabetical candidate priority; greedy retains its fixed 10^{-12} tie tolerance. The first two repetitions form $q = 2$, and all four form $q = 4$. The last two repetitions form a separate descriptive stability check, without another deployment arm.

The library identities and evaluation order are locked before any evaluation rollout. Each of the five logical conditions (Base, two selectors at each q) is evaluated twice on six tasks. Identical candidate sets share one execution condition within each model. Sonnet's $q = 2$ and $q = 4$ greedy sets coincide, reducing the maximum 120 evaluation executions to 108. Base is also shared across q. Repetition labels identify execution blocks, not matched provider random seeds.

For each model and selector, fitting separately on the first two and last two repetitions retains two common candidates out of four (Jaccard overlap $1/3$). Table 13 gives the deployed sets and fitted coverage. At $q = 4$, both selectors attain 0.375 on DeepSeek and 0.417 on Sonnet. Deployment differences alone therefore do not show that greedy found higher estimated coverage.

Endpoints and uncertainty. The primary endpoint is the $q = 4$ Greedy-minus-Singleton native reward difference, reported separately for each model. We average the two repetition differences within a task, then average over six tasks. Descriptive percentile 95% intervals use 10,000 whole-task bootstrap resamples with fixed seed 20260923 and shared resampling indices across contrasts. These intervals condition on the fitted libraries and this small task collection; they omit calibration-selection uncertainty and multiple-comparison correction. We also report the $q = 2$ contrast, changes between q settings, comparisons with Base, and task-level outcomes. All 108 evaluation rewards happen to be binary, so mean native reward equals full-credit success rate.

Table 13: Libraries fitted from fresh calibration. Candidates are listed alphabetically; $\hat{F}_n$ is fitted baseline-relative coverage. Coverage values across q use different estimated matrices. The selectors tie in fitted coverage at $q = 4$ for each model.

Model	q	Selector	Four retained candidates	$\hat{F}_n$
DeepSeek-V3.2	2	Greedy	browser-testing, docx, pptx, threejs	0.417
DeepSeek-V3.2	2	Singleton	dc-power-flow, docx, economic-dispatch, pptx	0.333
DeepSeek-V3.2	4	Greedy	browser-testing, docx, pptx, seismic-picker-selection	0.375
DeepSeek-V3.2	4	Singleton	docx, pptx, seismic-picker-selection, threejs	0.375
Sonnet	2	Greedy	browser-testing, dc-power-flow, docx, xlsx	0.583
Sonnet	2	Singleton	browser-testing, dc-power-flow, seisbench-model-api, xlsx	0.583
Sonnet	4	Greedy	browser-testing, dc-power-flow, docx, xlsx	0.417
Sonnet	4	Singleton	browser-testing, dc-power-flow, react-best-practices, xlsx	0.417

Table 14: Task-level full-credit counts for the primary $q = 4$ diagnostic, out of two executions. The Sonnet OCR row uses the uniform revised image policy. No task is omitted for a zero score or graded timeout.

Model	Task	Base	Greedy ($q = 4$)	Singleton ($q = 4$)
DeepSeek-V3.2	`invoice-fraud-detection`	1/2	1/2	0/2
DeepSeek-V3.2	`jpg-ocr-stat`	0/2	0/2	0/2
DeepSeek-V3.2	`organize-messy-files`	0/2	0/2	0/2
DeepSeek-V3.2	`paper-anonymizer`	0/2	0/2	0/2
DeepSeek-V3.2	`sales-pivot-analysis`	0/2	0/2	0/2
DeepSeek-V3.2	`xlsx-recover-data`	0/2	0/2	0/2
Sonnet	`invoice-fraud-detection`	1/2	1/2	0/2
Sonnet	`jpg-ocr-stat`	0/2	1/2	0/2
Sonnet	`organize-messy-files`	0/2	0/2	0/2
Sonnet	`paper-anonymizer`	2/2	2/2	2/2
Sonnet	`sales-pivot-analysis`	0/2	0/2	0/2
Sonnet	`xlsx-recover-data`	0/2	0/2	0/2

Table 3 reports all selector settings. The primary reward differences are $+0.083\,[0.000, 0.250]$ for DeepSeek and $+0.167\,[0.000, 0.333]$ for Sonnet. The $q = 2$ differences are $-0.083\,[-0.250, 0.000]$ and $+0.083\,[-0.167, 0.333]$, respectively. The change in the Greedy–Singleton contrast from $q = 2$ to $q = 4$ is $+0.167\,[0.000, 0.500]$ for DeepSeek and $+0.083\,[0.000, 0.250]$ for Sonnet. These comparisons change fitted library identities. In particular, Sonnet's greedy executions are shared across q, so its change comes entirely from the singleton comparison.

DeepSeek's primary advantage comes from invoice-fraud-detection; the other five tasks have zero success under both selectors. Sonnet's primary advantage comes from invoice-fraud-detection and jpg-ocr-stat. Both selectors succeed on both paper-anonymizer repetitions and fail on the remaining three tasks. Against Base, primary greedy has difference $0.000\,[0.000, 0.000]$ on DeepSeek and $+0.083\,[0.000, 0.250]$ on Sonnet. The zero empirical interval on DeepSeek reflects identical observed task means and does not establish population equivalence. The positive primary means do not establish stable superiority or a general benefit from increasing q.

Execution and exposure. Every retained execution passes checks of task and model identity, source hashes, assigned candidates, original instructions, the 64-entry background catalog and absence of a preload prompt. Fourteen evaluation executions end in agent timeouts with native verifier scores; their outcomes remain included. The tool records contain 13 explicit skill invocations, all in selector conditions. DeepSeek invokes the background `pdflatex` skill once; Sonnet invokes `xlsx` twelve times on sales-pivot-analysis and xlsx-recover-data. None of these calls occurs on the invoice or OCR tasks that contribute Sonnet's primary advantage. Six executions contain explicit skill-path mentions in tool input/display metadata. These observations do not identify correct

Table 15: Sonnet OCR outcomes before and after uniform image repair. Counts use only native-scored executions. One original $q = 4$ singleton execution remained unscored after API rejections; it is not imputed. All four conditions were repeated under the revised policy.

Condition	Original scored	Original unscored	Revised scored
Base	1/2	0	0/2
Greedy ($q = 2, 4$)	0/2	0	1/2
Singleton ($q = 2$)	0/2	0	0/2
Singleton ($q = 4$)	0/1	1	0/2

skill use as the cause of a reward difference; the counters do not fully capture ordinary file reads or influence from catalog descriptions.

Background and verifier recovery. Three documented operational amendments accompany the experiment. First, an upstream background-catalog change invalidated five calibration configurations. Their original scores and traces remain archived; the configurations were rerun under a fixed background-source revision before fitting. The other 355 scored configurations present at that pause were retained. Earlier catalog hashes match the target entries, but their background skill bodies had used a mutable upstream revision; the later pin does not retroactively establish identical bodies.

Second, two calibration verifiers failed during dependency installation before running their tests. The unchanged native tests were rerun offline against byte-identical archived agent workbooks and CSVs, with no further agent execution. Two preselected control outputs matched all 27 individual native test results before the affected outputs were graded. Both recovered outcomes are zero. The recovery records preserve the separate grading environment and its Python patch/base-system differences.

Uniform OCR image repair. After 41 evaluation scores were available, the Sonnet OCR $q = 4$ singleton condition exhausted three unscored attempts because the provider rejected images larger than 2,000 pixels in a many-image request. Resizing could alter OCR accuracy, so the repair repeats all eight Sonnet OCR executions under one image policy, including the seven already scored executions, independently of reward. The fitted libraries, original task files, instructions, verifier and time limits remain fixed; calibration is not repeated. The primary comparison uses all eight revised outcomes, while Table 15 retains the original partial cohort separately.

Only this model/task combination uses the revised proxy. Compatible inline image bytes are unchanged. Oversized images retain orientation and aspect ratio, are resized with Pillow LANCZOS to a maximum side of 2,000 pixels, and are encoded as PNG. A fixed 3,500,000-byte bound triggers further proportional 80% size reductions if needed. No image is cropped, removed or reordered, and non-image request content is preserved. Logs retain original and submitted image hashes, sizes and transformation records. The policy and isolated runtime were frozen before the revised executions, but after outcomes from the original evaluation were known. Original and revised OCR observations are not interchangeable.

Stopping and accounting. The first valid native score, including zero, partial credit or a graded timeout, completes a cell. Unscored attempts have a fixed three-attempt ceiling; an exhausted cell or configuration mismatch stops new launches. The revised OCR cells have separate attempt directories and the same ceiling, with the original attempts preserved. No tasks, models or libraries were substituted based on evaluation scores. Final checks verify all 876 retained records and their raw artifact hashes. The campaign records approximately $1,557.08 in measured or estimated model-call usage, including failed attempts, access checks and both OCR configurations. Two task attempts and two access checks have unknown costs. The estimate excludes host compute and other campaigns and is not a provider invoice.

H MCPAGENTBENCH REDUNDANCY AND CALIBRATION SUBSAMPLING

The MCPAgentBench study compares selection rules on released score matrices. The primary protocol chooses the baseline using calibration runs only; the sensitivity analysis chooses it using all runs. Both evaluate coverage on separate rollout scores for the same tasks. The six partitions share runs, so their mean and SE are descriptive summaries rather than estimates from independent replicates.

Table 16 reports held-out coverage across all evaluated budgets. Table 17 gives the corresponding coverage gaps for the primary and sensitivity protocols. For $k \leq 5$, the reference is the exhaustive held-out optimum. Rows marked † use greedy on the held-out matrix; their reported gaps are lower bounds on the gaps to the optimum. Table 18 reports the single-setting control ($m = 11$), with an exhaustive optimum at every budget.

Table 16: MCPAgentBench held-out coverage with 30 model–setting candidates (mean ± SE over the 6 rollout splits). The baseline is chosen on calibration runs. Raw-success greedy selects identical libraries here because no cost term is used and is omitted.

k	Greedy	Singleton	Diversity	Random	Advantage	Adv > 0
2	0.203 ± 0.009	0.185 ± 0.013	0.175 ± 0.009	0.152 ± 0.008	$+0.018 \pm 0.006$	5/6
3	0.215 ± 0.010	0.205 ± 0.011	0.180 ± 0.009	0.177 ± 0.007	$+0.010 \pm 0.006$	5/6
4	0.224 ± 0.011	0.217 ± 0.008	0.193 ± 0.010	0.196 ± 0.009	$+0.006 \pm 0.004$	4/6
5	0.229 ± 0.011	0.228 ± 0.009	0.202 ± 0.008	0.209 ± 0.009	$+0.002 \pm 0.003$	4/6
6	0.234 ± 0.010	0.232 ± 0.009	0.207 ± 0.010	0.216 ± 0.008	$+0.002 \pm 0.002$	5/6
7	0.237 ± 0.010	0.239 ± 0.009	0.226 ± 0.009	0.223 ± 0.008	-0.002 ± 0.002	1/6
8	0.240 ± 0.010	0.241 ± 0.008	0.237 ± 0.010	0.229 ± 0.009	-0.002 ± 0.002	1/6

Routing with coarse descriptors. For nearest-winner routing with coarse category descriptors, winner agreement falls from 0.51 at $k = 2$ to 0.42 at $k = 8$. This diagnostic motivates the extra descriptor assumptions in Theorem 3; it does not establish a general deployment guarantee.

H.1 TASK SUBSAMPLING BY IDENTIFIER GROUP

We group tasks by strings parsed from their MCPAgentBench identifiers: $G = 5$ groups (`day`: 30 tasks, `pro`: 32, `parallel`: 40, `sequential`: 40, `multi3`: 38). These labels are syntactic groups rather than semantic annotations. We use all 6 rollout splits and choose each split's baseline on calibration runs, as in Section 4.4.

The full arm runs greedy on the complete 180×30 calibration matrix. The subsampled arm takes s tasks per group, averages their scores into a $G \times m$ matrix, and runs greedy weighted by the groups' frequencies. Results average over 10 subsample draws per split. Both arms are evaluated on the full 180-task held-out matrix. The comparison includes both subsampling and averaging; the "all" arm averages within groups using every task.

At $s = 2$, calibration uses 10 tasks: 5.6% of the matrix rows. With m and q unchanged, this would reduce calibration rollouts by $18\times$. The held-out coverage loss is $+0.009$–$+0.027$ (roughly 4–13% of the full-arm value), positive in 6/6 splits at $k \leq 6$ and 5/6 at $k = 8$. The loss shrinks monotonically as s grows; averaging with all rows gives losses between -0.005 and $+0.001$.

The measured within-group dispersion averages 0.17 across (group, skill) cells, but the worst cell reaches 0.42. A worst-case bound based on that dispersion would be loose, as in the descriptor-insufficiency analysis associated with Theorem 3. The study therefore reports a retrospective calibration-budget comparison; a sharp guarantee for group averaging remains future work.

The matrix was already collected, so the $18\times$ figure is a simulated acquisition budget, not a measured live cost saving. The syntactic groups also give no demonstrated robustness benefit. Under the same protocol, leave-one-group-out transfer reaches 0.73–1.03 of the within-group oracle at $k \geq 4$ across the four non-trivial groups. It is weakest for `pro`; the `day` group has essentially identical coverage under any library. Companion selection variants using group frequencies, per-group quotas or robust max–min objectives give no held-out gains over plain greedy.

Table 17: MCPAgentBench coverage gaps to the held-out reference (mean ± SE over the 6 rollout splits). Ratio is singleton gap divided by greedy gap. †: the reference uses greedy on the held-out matrix in place of exhaustive optimization.

	Primary (calibration baseline)			Sensitivity (baseline from all runs)		
k	Greedy	Singleton	Ratio	Greedy	Singleton	Ratio
2	0.006 ± 0.002	0.024 ± 0.005	$4.1\times$	0.006 ± 0.003	0.038 ± 0.007	$6.7\times$
3	0.013 ± 0.002	0.023 ± 0.005	$1.8\times$	0.015 ± 0.002	0.025 ± 0.001	$1.7\times$
4	0.016 ± 0.003	0.023 ± 0.002	$1.4\times$	0.020 ± 0.002	0.027 ± 0.004	$1.3\times$
5	0.017 ± 0.003	0.019 ± 0.003	$1.1\times$	0.019 ± 0.002	0.025 ± 0.006	$1.3\times$
$6^\dagger$	0.016 ± 0.002	0.019 ± 0.003	$1.2\times$	0.019 ± 0.002	0.022 ± 0.003	$1.1\times$
$7^\dagger$	0.018 ± 0.002	0.016 ± 0.002	$0.9\times$	0.021 ± 0.002	0.021 ± 0.002	$1.0\times$
$8^\dagger$	0.018 ± 0.002	0.017 ± 0.002	$0.9\times$	0.020 ± 0.002	0.023 ± 0.003	$1.1\times$

Table 18: Single-setting control pool ($m = 11$), with the baseline chosen on calibration runs and exhaustive optimization at every k. Entries are mean ± SE over the 6 rollout splits.

k	Advantage	Adv > 0	Greedy gap	Singleton gap
2	-0.002 ± 0.004	1/6	0.005 ± 0.004	0.003 ± 0.003
3	$+0.005 \pm 0.002$	5/6	0.005 ± 0.002	0.010 ± 0.001
4	$+0.011 \pm 0.002$	6/6	0.006 ± 0.002	0.017 ± 0.001
5	$+0.009 \pm 0.002$	6/6	0.008 ± 0.001	0.017 ± 0.001
6	$+0.006 \pm 0.002$	6/6	0.009 ± 0.001	0.016 ± 0.002
7	$+0.006 \pm 0.003$	5/6	0.009 ± 0.002	0.015 ± 0.003
8	-0.000 ± 0.002	2/6	0.007 ± 0.002	0.007 ± 0.003

Table 19: Held-out coverage loss from task subsampling and group averaging (full-matrix greedy minus greedy on group means; mean ± SE over the 6 rollout splits, 10 subsample draws per split). Simulated calibration budgets: $s = 2$ uses 5.6% of the rows ($18\times$ fewer rollouts), $s = 4$ uses 11.1%, and $s = 8$ uses 22.2%. The "all" arm forms a $G \times m$ matrix using every task.

k	$s = 2$ (5.6%)	$s = 4$ (11.1%)	$s = 8$ (22.2%)	all (100%)
2	$+0.027 \pm 0.002$	$+0.022 \pm 0.003$	$+0.014 \pm 0.003$	-0.001 ± 0.001
3	$+0.023 \pm 0.003$	$+0.013 \pm 0.002$	$+0.007 \pm 0.002$	-0.005 ± 0.003
4	$+0.022 \pm 0.004$	$+0.014 \pm 0.003$	$+0.006 \pm 0.001$	-0.005 ± 0.002
5	$+0.016 \pm 0.003$	$+0.013 \pm 0.003$	$+0.009 \pm 0.002$	-0.001 ± 0.003
6	$+0.014 \pm 0.002$	$+0.012 \pm 0.001$	$+0.006 \pm 0.001$	$+0.001 \pm 0.002$
8	$+0.009 \pm 0.002$	$+0.006 \pm 0.002$	$+0.003 \pm 0.002$	$+0.000 \pm 0.003$

I SYNTHETIC EXPERIMENTS AND DIAGNOSTICS

The synthetic studies examine redundancy, calibration error and routing under known utilities. We first give the coverage and scaling summaries from Section 4, then describe perturbation checks and tests that remove individual assumptions. Some legacy diagnostics retain only plots and summaries; Appendix K records the available evidence and uncertainty limits.

I.1 SYNTHETIC COVERAGE AND CALIBRATION SUMMARIES

Table 20 summarizes the synthetic results reported in Section 4.

Figure 4 shows the redundant-clone ratios across k: singleton/OPT approaches $1/k$, while greedy remains optimal.

Figure 5 shows the qualitative n- and q-scaling fits for Theorem 2.

I.2 DIAGNOSTIC QUANTITIES

The supplementary plots use the archived auxiliary-selector names *top-k marginal*, *stochastic lazy greedy* and *local search*. Some summaries use *greedy-family* or *singleton-like*. They do not establish which implementations coincide with the primary selectors in Section 2.

The entrywise perturbation size and the induced coverage perturbation of a fixed library are

$$\alpha = \max_{i,\ e \in V \cup \{0\}} |\tilde{u}_{i,e} - u_{i,e}|\,, \qquad \Delta_{\mathrm{stab}}(S) = |F_{\tilde{U}}(S) - F_U(S)|\,,$$

and Lemma 5 predicts $\Delta_{\mathrm{stab}}(S) \le 2\alpha$.

The uniform objective discrepancy in Lemma 6 and the coverage gap of greedy fitted to the perturbed objective are

$$\varepsilon = \sup_{|S| \le k} |G(S) - H(S)|\,, \qquad \mathrm{Gap}_{\mathrm{transfer}} = \max_{|S| \le k} G(S) - G(S_g^H).$$

The routing regret used for Theorem 3, the optimism gap used for the leakage stress test, and the sampled diminishing-returns violation rate used in the interaction stress test are

$$\mathrm{Regret}_{\mathrm{route}} = F_v(S) - J_v(\rho, S), \qquad \mathrm{Optimism} = F_{\mathrm{cal}}(S) - F_{\mathrm{test}}(S),$$
$$\mathrm{ViolationRate} = \Pr[\Delta(e \mid A) < \Delta(e \mid B)]\,.$$

At $\alpha = 0.10$, Lemma 5 gives $2\alpha = 0.20$. The observed mean coverage deviation was $\overline{\Delta}_{\mathrm{stab}} \approx 0.01$, giving a ratio to the bound of $\frac{\overline{\Delta}_{\mathrm{stab}}}{2\alpha} \approx 0.05$.

In the routing diagnostic, mean regret rose from $\mathrm{Regret}_{\mathrm{clean}} = 0.013$ with clean descriptors to $\mathrm{Regret}_{\mathrm{random\ embed}} = 0.178$ with random embeddings, a factor of $\frac{\mathrm{Regret}_{\mathrm{random\ embed}}}{\mathrm{Regret}_{\mathrm{clean}}} \approx 13.7$.

Table 21 relates the appendix diagnostics to the formal results.

I.3 MAX-STABILITY PERTURBATION CHECKS

Lemma 5 bounds the change in coverage when every matrix entry changes by at most a specified amount. We perturb the full matrix directly and measure that change.

At each perturbation level, we apply the same corruption rule to skill and baseline cells,

$$\tilde{u}_{i,e} = \mathrm{clip}(u_{i,e} + \xi_{i,e},\, 0,\, 1)\,, \qquad \tilde{u}_{i,0} = \mathrm{clip}(u_{i,0} + \xi_{i,0},\, 0,\, 1)\,,$$

where $|\xi_{i,e}| \le \alpha$.

For a fixed perturbation level, we report the mean absolute coverage deviation and the normalized slack ratio

$$\overline{\Delta}_{\mathrm{stab}}(\alpha) = \mathbb{E}[\Delta_{\mathrm{stab}}(S)]\,, \qquad R_{\mathrm{stab}}(\alpha) = \frac{\overline{\Delta}_{\mathrm{stab}}(\alpha)}{2\alpha}.$$

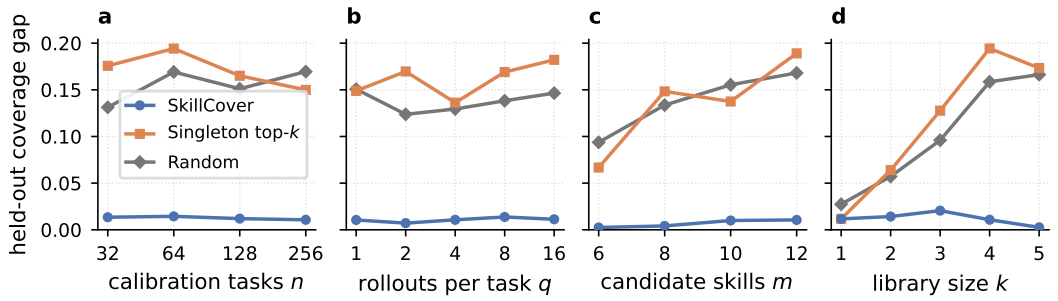


Figure 3: Held-out oracle sweeps on synthetic latent full-matrix worlds, aggregated over 30 seeds. SkillCover stays close to the held-out oracle across budget and calibration sweeps, whereas singleton ranking remains substantially more redundant.

Table 20: Synthetic coverage, calibration and routing summaries. Descriptor corruption is evaluated as a stress test; the scaling fits use Gaussian cell noise in one path.

Setting	Summary statistic
Redundant clones	Greedy/OPT $= 1.000$. Worst singleton/OPT $= 0.2052$ at $k = 5$, $q/p = 0.98$.
Latent held-out sweeps	$\overline{\text{Gap}}_{\text{SC}} = 0.0106$. Mean advantage $= 0.1340$ over singleton and 0.1208 over random. At $k = 4$: 0.0108 for SkillCover versus 0.1943 for singleton.
Theorem 2 scaling	Observed log-slope $\hat{b}_n \approx -0.40$. Observed log-slope $\hat{b}_q \approx -0.50$. Spearman correlations 0.71 for n and 0.83 for q.
Conditional routing	Clean descriptors give regret 0.013. Corrupted descriptors give regret 0.178 (random embedding), 0.135 (missing variable), and 0.074 (aliased).

Across the sweep $\alpha \in \{0,\ 0.05,\ 0.10,\ 0.20\}$, the observed deviation grows approximately linearly with α. Lemma 5 supplies an upper bound rather than predicting that curve. At nonzero perturbation levels, the reported ratio is approximately $R_{\text{stab}}(\alpha) \approx 0.05$.

At $\alpha = 0.20$, the lemma gives $2\alpha = 0.40$. The observed mean deviation is $\overline{\Delta}_{\text{stab}}(0.20) \approx 0.02$, so the slack ratio remains $R_{\text{stab}}(0.20) \approx 0.05$.

Table 22 reports the audit at this perturbation level. The measured deviations are nearly identical across the auxiliary selectors. This similarity is an empirical observation; Lemma 5 provides the same upper bound for every library.

These measurements assess numerical slack; the lemma holds deterministically under the stated perturbation condition.

I.4 Greedy-transfer checks on enumerated worlds

Lemma 6 transfers the greedy approximation guarantee from a perturbed objective to the target objective through a uniform error bound. The diagnostic uses small synthetic worlds whose target optimum can be enumerated exactly.

Let $G(S)$ be the true objective and $H(S)$ the perturbed objective, with perturbation size $\varepsilon = \sup_{|S|\le k} |G(S) - H(S)|$. Greedy on the perturbed objective returns S_g^H, whose coverage gap under the true objective is $\text{Gap}_{\text{transfer}} = \max_{|S|\le k} G(S) - G(S_g^H)$.

For entrywise matrix noise of size α, Lemma 5 gives $\varepsilon \le 2\alpha$, and Lemma 6 yields the envelope

$$\text{Gap}_{\text{transfer}} \le \left(2 - \frac{1}{e}\right)\varepsilon \le 2\left(2 - \frac{1}{e}\right)\alpha, \qquad 2\left(2 - \frac{1}{e}\right) \approx 3.26.$$

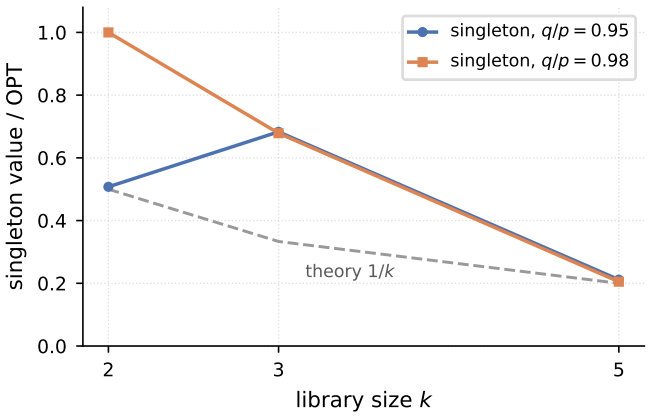


Figure 4: Redundant-clone study based on Theorem 1. The singleton/OPT ratio approaches the predicted $1/k$ behavior, while greedy remains optimal.

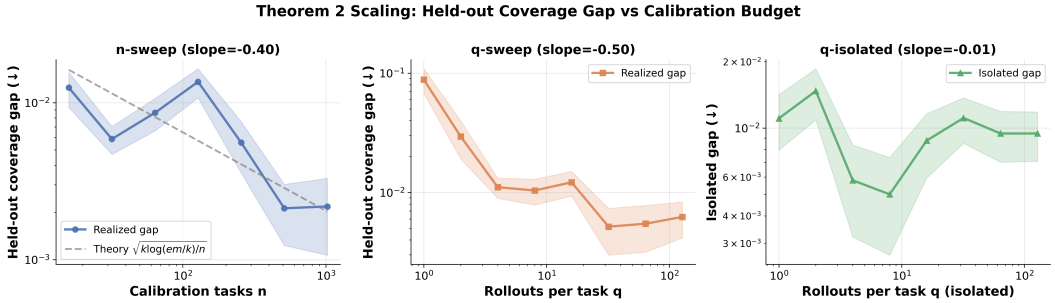


Figure 5: Qualitative scaling check for Theorem 2. The n- and q-panels show held-out regret decreasing in the directions predicted by the sampling terms. One harness uses Gaussian cell noise, so the comparison concerns scaling trends rather than the literal rollout model.

At $\alpha = 0.20$, the transfer envelope is $2\left(2 - \frac{1}{e}\right)\alpha \approx 0.652$. In the enumerated-world sweep, the empirical greedy transfer gap is $\mathrm{Gap}_{\mathrm{transfer}} \approx 0$.

Across the α-sweep, the observed SkillCover coverage gap stays essentially zero, while the auxiliary selectors have nonzero gaps. Table 22 reports this comparison alongside the stability audit. Theorem 2 uses the chain $\hat{F}_n \longrightarrow F_n \longrightarrow F$; these small-world results describe the transfer step without measuring every source of calibration error.

I.5 Conditional routing diagnostics

With the selected library fixed as required by Theorem 3, routed regret is $\mathrm{Regret}_{\mathrm{route}} = F_v(S) - J_v(\rho, S)$. The clean Lipschitz experiment varies calibration density and nearest-neighbor radius in synthetic descriptor worlds while keeping the descriptor map informative. The empirical bound slack ratio is

$$R_{\mathrm{route}} = \frac{\mathrm{Regret}_{\mathrm{route}}}{2L\,\mathbb{E}[\min_i d(X, x_i)] + 2\epsilon_{\mathrm{desc}}},$$

and across the clean sweep this ratio stayed in the rough range $R_{\mathrm{route}} \in [0.01,\ 0.05]$.

We then corrupt the descriptors while leaving the library fixed to examine sensitivity to descriptor quality.

Table 21: Appendix diagnostics. "In-regime" experiments target assumptions or approximation terms in Lemmas 5–6 and Theorems 2–3. "Out-of-regime" experiments violate one assumption to test its role.

Suite	Quantity	Regime	Empirical pattern	Supports
Max-stability perturbation	$\Delta_{\mathrm{stab}}(S)$ vs. α	In-regime	Linear growth, roughly 5% of the 2α bound	Lemma 5
Greedy transfer	$\mathrm{Gap}_{\mathrm{transfer}}$ vs. ε	In-regime	Greedy gap essentially zero across the perturbation sweep; bound very loose	Lemma 6
Clean Lipschitz routing	$\mathrm{Regret}_{\mathrm{route}}$ vs. NN radius	In-regime	Regret is only 1%–5% of the theorem envelope	Theorem 3
Descriptor corruption	$\mathrm{Regret}_{\mathrm{route}}$ under corruption	Out-of-regime	Clean descriptors are low-regret; corruption increases regret by $5\times$–$14\times$	Failure boundary for Theorem 3
Adaptive leakage	$\mathrm{Optimism}$	Out-of-regime	Calibration optimism increases sharply when the pool depends on calibration failures	Failure boundary for Theorem 2
Interaction stress	$\mathrm{ViolationRate}$ and greedy/OPT	Out-of-regime	Submodularity violations appear, but greedy degradation is modest	Failure boundary for Theorem 1
Heavy-tail and AR(1) noise	held-out gap	Out-of-regime	Greedy remains stable; singleton-like ranking is more fragile	Failure boundary for Theorem 2

Table 22: Perturbation diagnostics for Lemma 5 (max-stability) and Lemma 6 (greedy transfer) at $\alpha = 0.20$. Max-stability perturbs every entry, including the baseline, by at most α; transfer evaluates greedy fitted to the perturbed objective under the target objective. Both use the stated theorem conditions. Deviations are far below the bounds, and max-stability deviation grows approximately linearly across $\alpha \in \{0,\, 0.05,\, 0.10,\, 0.20\}$.

Audit	α	Theory envelope	Observed	Slack ratio
Max-stability (Lemma 5)	0.20	$2\alpha = 0.40$	$\overline{\Delta}_{\mathrm{stab}} \approx 0.02$	≈ 0.05
Greedy transfer (Lemma 6)	0.20	$2(2 - 1/e)\,\alpha \approx 0.652$	$\mathrm{Gap}_{\mathrm{transfer}} \approx 0$	≈ 0

The mean routed regrets under clean descriptors, random descriptor embeddings, a missing-variable descriptor, and descriptor aliasing were

$$\mathrm{Regret}_{\mathrm{clean}} = 0.013, \qquad \mathrm{Regret}_{\mathrm{random\ embed}} = 0.178,$$
$$\mathrm{Regret}_{\mathrm{missing\ var}} = 0.135, \qquad \mathrm{Regret}_{\mathrm{aliased}} = 0.074.$$

Table 23: Routing diagnostics associated with Theorem 3. The clean row uses informative descriptors; the other rows corrupt the descriptor while holding the library fixed.

Descriptor mode	Mean routed regret	Relative to clean	Regime status
Clean	0.013	1.0×	In-regime
Random embedding	0.178	13.7×	Out-of-regime
Missing variable	0.135	10.4×	Out-of-regime
Aliased descriptor	0.074	5.7×	Out-of-regime

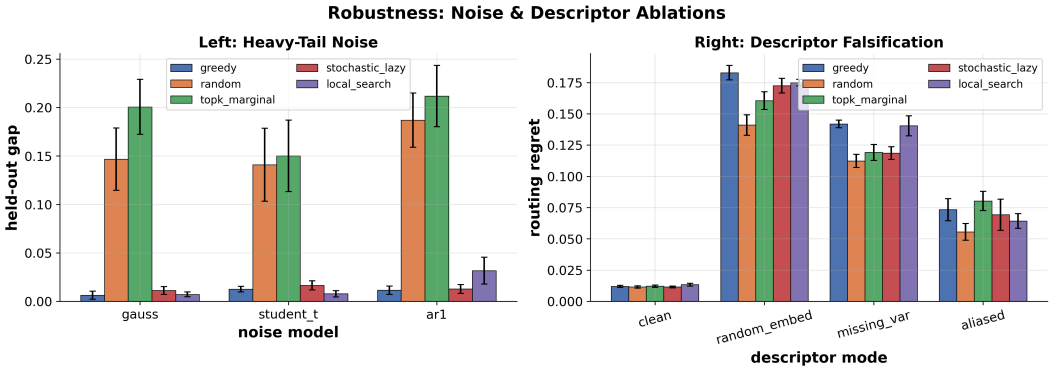


Figure 6: Routing and noise diagnostics. Right: descriptor corruption increases reported routing regret for a fixed library. Left: heavy-tailed and AR(1) perturbations produce larger coverage gaps for the singleton-like control than for greedy in these synthetic worlds. The panels examine sensitivity to the conditions in Theorems 3 and 2.

Relative to the clean world, the corruption multipliers are

$$\frac{\mathrm{Regret}_{\text{random embed}}}{\mathrm{Regret}_{\text{clean}}} \approx 13.7,$$
$$\frac{\mathrm{Regret}_{\text{missing var}}}{\mathrm{Regret}_{\text{clean}}} \approx 10.4,$$
$$\frac{\mathrm{Regret}_{\text{aliased}}}{\mathrm{Regret}_{\text{clean}}} \approx 5.7.$$

Table 23 summarizes the reported regrets. The corruption tests examine descriptors that omit or obscure information about the best skill. Their interpretation depends on the distinction between task-level and descriptor-conditional coverage in Appendix E.

Figure 6 shows the descriptor-corruption and noise results.

I.6 Sensitivity to pool construction, interactions and noise

The following tests change pool construction, skill interactions or the noise model to examine the limits of the selection analysis.

Adaptive leakage. Theorem 2 assumes that the candidate pool and tie-breaking rule are fixed independently of calibration outcomes. The leakage test allows the pool to depend on calibration failures, violating that condition.

The optimism metric is $\mathrm{Optimism} = F_{\mathrm{cal}}(S) - F_{\mathrm{test}}(S)$. For SkillCover greedy under a clean split, the mean optimism was $\mathrm{Optimism}_{\mathrm{clean}} \approx 0.007$; when the candidate pool was allowed to depend on calibration failures, it became $\mathrm{Optimism}_{\mathrm{leak}} \approx 0.029$, an inflation factor of $\frac{\mathrm{Optimism}_{\mathrm{leak}}}{\mathrm{Optimism}_{\mathrm{clean}}} \approx 4.1$.

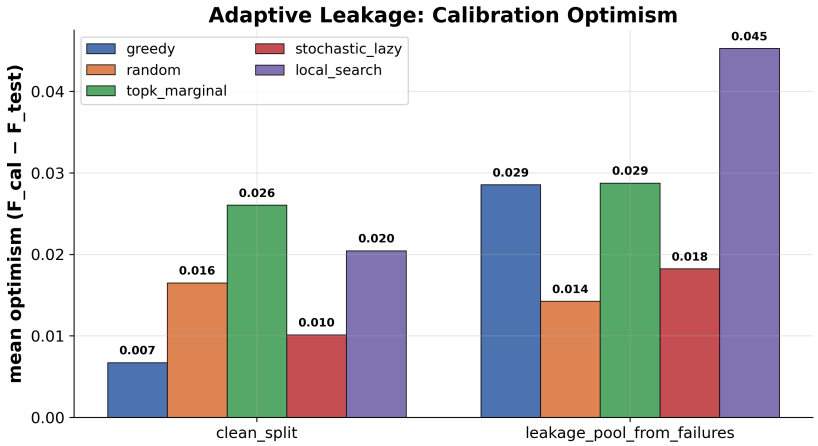


Figure 7: Adaptive leakage test. The candidate pool depends on calibration failures, violating the fixed-pool assumption of Theorem 2. Greedy's calibration-to-test optimism increases under this dependence.

Figure 7 shows the increase in optimism. The finite-class guarantee does not apply as stated when the pool depends on calibration outcomes; Proposition 9 gives a counterexample.

AND-gate interactions. Theorem 1 relies on the pointwise-maximum one-skill-per-task model. The interaction tests add tasks that require two skills together.

The sampled diminishing-returns violation rate rises to $\mathrm{ViolationRate} \approx 0.07$ in the stronger stress settings. Even then, the observed greedy approximation ratio remained in the rough range $\frac{F(S_g)}{F(S^\star)} \in [0.92,\ 1.00]$ in the tested worlds.

Greedy performs well in these particular interaction tests. That observation does not extend Theorem 1; Remark 3 shows that greedy can fail arbitrarily badly with interactions.

Heavy-tail and correlated rollout noise. Theorem 2 requires conditionally unbiased cell estimates with sub-Gaussian error decreasing at the stated rate in q. We examine noise sensitivity using Student-t and AR(1)-correlated perturbations.

The greedy-family methods had gaps in the rough range $\mathrm{Gap}_{\text{greedy family}} \in [0.007,\ 0.03]$, while the singleton-like top-k control reached $\mathrm{Gap}_{\text{top-k, AR(1)}} \approx 0.21$ under AR(1) noise.

These ranges describe performance under the tested perturbations. They do not establish the concentration assumptions or rollout rate required by the theorem.

I.7 Interpretation and estimator scope

The archived perturbation, transfer and clean-descriptor diagnostics report deviations that are small relative to their stated comparison terms:

$$\Delta_{\mathrm{stab}}(S) \ll 2\alpha, \qquad \mathrm{Gap}_{\mathrm{transfer}} \ll \left(2 - \frac{1}{e}\right)\varepsilon,$$
$$\mathrm{Regret}_{\mathrm{route}} \ll 2L\,\mathbb{E}[\min_i d(X, x_i)] + 2\epsilon_{\mathrm{desc}}.$$

The descriptor-corruption, leakage and interaction diagnostics report

$$\mathrm{Regret}_{\text{corrupted desc}} \gg \mathrm{Regret}_{\mathrm{clean}},$$
$$\mathrm{Optimism}_{\mathrm{leak}} > \mathrm{Optimism}_{\mathrm{clean}},$$
$$\mathrm{ViolationRate}_{\mathrm{interaction}} > 0.$$

One scaling harness adds Gaussian noise to cell means. Its n- and q-sweeps check qualitative trends for Theorem 2, not the exact repeated-rollout model. The fresh-calibration diagnostic in Appendix G.5 uses real repeated rollouts at $q = 2, 4$, but evaluates fitted libraries and deployment rather than the bound's error constants.

The repeated-rollout estimator is $\hat{u}_{i,e} = \frac{1}{q}\sum_{r=1}^{q} Y_{i,e,r}$. The Gaussian surrogate instead uses

$$\hat{u}_{i,e} = \mathrm{clip}(u_e(Z_i) + \varepsilon_{i,e},\, 0,\, 1)\,, \qquad \varepsilon_{i,e} \sim \mathcal{N}\left(0,\, \frac{\sigma^2}{q}\right).$$

Perturbing and clipping a cell mean gives a different estimator from averaging rollout scores. The deterministic perturbation checks and the tests of pool construction, interactions and descriptor corruption examine other parts of the analysis.

J Implementation Details

This appendix describes the calibration matrices, selection code and synthetic generators used by the studies. The matrix stores mean clipped utilities, including a separate baseline column. Baseline subtraction occurs when computing coverage; greedy selection then uses the stored matrix without additional rollouts.

For repeated rollouts, the calibration tensor is $\mathcal{Y}^{\mathrm{cal}} \in [0,1]^{n\times(m+1)\times q}$ and its cell-mean matrix is $\hat{U}^{\mathrm{cal}} \in [0,1]^{n\times(m+1)}$. The held-out tensor is $\mathcal{Y}^{\mathrm{test}} \in [0,1]^{n_{\mathrm{test}}\times(m+1)\times q}$, with cell-mean matrix $\hat{U}^{\mathrm{test}} \in [0,1]^{n_{\mathrm{test}}\times(m+1)}$.

The calibration objective, held-out coverage and routed value are computed separately:

$$\hat{F}_n(S) = \frac{1}{n}\sum_{i=1}^{n}\left[\max_{e\in S\cup\{0\}} \hat{u}_{i,e} - \hat{u}_{i,0}\right],$$

$$F_{\mathrm{holdout}}(S) = \frac{1}{n_{\mathrm{test}}}\sum_{i=1}^{n_{\mathrm{test}}}\left[\max_{e\in S\cup\{0\}} u^{\mathrm{test}}_{i,e} - u^{\mathrm{test}}_{i,0}\right],$$

$$J_{\mathrm{holdout}}(\rho, S) = \frac{1}{n_{\mathrm{test}}}\sum_{i=1}^{n_{\mathrm{test}}}\left[u^{\mathrm{test}}_{i,\rho(x_i)} - u^{\mathrm{test}}_{i,0}\right].$$

Here $u^{\mathrm{test}}_{i,e}$ denotes the utility matrix used for evaluation: known utilities in synthetic oracle studies and observed held-out scores in real-matrix studies. The archive uses $F_{\mathrm{holdout}}(S)$ for held-out coverage, corresponding to $\hat{F}_{\mathrm{hold}}(S)$ for the observed real-data metric in Section 4. Calibration coverage $\hat{F}_n(S)$, held-out coverage $F_{\mathrm{holdout}}(S)$, and routed value $J_{\mathrm{holdout}}(\rho, S)$ are recorded separately.

J.1 Code paths and package structure

The real-benchmark scripts are `experiments/real_benchmark_unique_splits.py` for MCPAgentBench's six-split comparison, `experiments/type_level_clean.py` for task subsampling, and `experiments/skillsbench_matrix_runner.py` plus `experiments/skillsbench_reporting_export.py` for the historical real-skill matrix. The fresh-calibration protocol and fitting functions are in `experiments/skillsbench_calibration_validation.py`; separate adapters record its operational amendments. The table generator `reporting/tables/skillsbench_calibration.py` verifies the completed bundle and regenerates its five tables.

We propose organizing reusable components under `src/skillcover/`, with responsibilities listed in Table 25. The selector has no trainable parameters; for a given pool and budget, its inputs are the calibration matrix and deterministic tie rule.

The repeated-rollout path computes $\hat{u}_{i,e} = \frac{1}{q}\sum_{r=1}^{q} Y_{i,e,r}$; the surrogate-noise path computes

$$\hat{u}_{i,e} = \mathrm{clip}(u_e(Z_i) + \varepsilon_{i,e},\, 0,\, 1)\,, \qquad \varepsilon_{i,e} \sim \mathcal{N}\left(0,\, \frac{\sigma^2}{q}\right).$$

Table 24: Synthetic code paths. The first two rows supply the reported synthetic evidence. The final row was inspected as a sandbox and is not treated as the paper implementation.

Code path	Main role	Noise model	Status in this paper
Stage 1 preliminary pipeline	Full-matrix selection baseline and latent-world sweeps	Literal repeated Bernoulli rollouts averaged over q	Used for synthetic fixed-matrix evidence
Stage 3 research harness	Theorem 2 and Theorem 3 synthetic studies	Effective Gaussian cell noise scaled by $1/\sqrt{q}$ in one path	Used only as qualitative theorem-consistency evidence
Stage 4 ablation sandbox	Alternative selectors and stress branches	Mixed synthetic perturbation models	Not treated as the paper implementation

Table 25: Proposed module organization for the SkillCover artifact. This is a layout plan, not an inventory of the executed scripts.

Module	Responsibility
`objective.py`	Clipped utility transformation, baseline-relative coverage objective, held-out oracle evaluation
`selectors.py`	Greedy SkillCover, singleton ranking, random library, exhaustive OPT for small m
`estimators.py`	Literal rollout averaging, surrogate cell-noise estimator, split construction
`routing.py`	Nearest-winner router, descriptor corruption modes, routing regret computation
`synthetic.py`	Redundant-clone worlds, latent-region worlds, clean Lipschitz descriptor worlds
`diagnostics.py`	Max-stability, greedy-transfer, leakage, interaction, and noise stress suites
`io.py`	Saving matrices, subsets, seeds, and per-run summaries
`results.py`	Aggregation across seeds and generation of paper tables
`plotting.py`	Figure scripts only; no experimental logic

The literal path uses the estimator in Theorem 2. The surrogate path is used in part of the Stage 3 scaling experiments and checks only qualitative trends.

For example, the literal path with $q = 4$ and rollout utilities $(Y_{i,e,1}, Y_{i,e,2}, Y_{i,e,3}, Y_{i,e,4}) = (1, 0, 1, 1)$ gives $\hat{u}_{i,e} = \frac{1+0+1+1}{4} = 0.75$. The surrogate path with $u_e(Z_i) = 0.72$ and perturbation $\varepsilon_{i,e} = -0.05$ instead stores $\hat{u}_{i,e} = \mathrm{clip}(0.72 - 0.05, 0, 1) = 0.67$.

Table 24 identifies the synthetic code paths by their archived stage names.

Table 25 records the proposed responsibilities for calibration, selection and routing code.

J.2 Logged quantities and normalization audit

The raw score before clipping follows the utility definition in Appendix A. For each task i, candidate e, and rollout r, the stored raw score and the stored clipped utility are

$$R_{i,e,r} = r_{i,e,r} - c^{\mathrm{step}}_{i,e,r} - c^{\mathrm{token}}_{i,e,r} - c^{\mathrm{api}}_{i,e,r} - c^{\mathrm{tool}}_{i,e,r} - c^{\mathrm{invalid}}_{i,e,r} - c^{\mathrm{skill}}_{e},$$
$$Y_{i,e,r} = \mathrm{clip}(aR_{i,e,r} + b,\, 0,\, 1)\,.$$

The baseline-relative cell contribution, computed only after clipping, is $\Delta_{i,e} = \hat{u}_{i,e} - \hat{u}_{i,0}$. Linearity of averaging gives the identity

$$\Delta_{i,e} = \frac{1}{q}\sum_{r=1}^{q}(Y_{i,e,r} - Y_{i,0,r})\,.$$

Table 26: Rollout and matrix fields required for an auditable SkillCover run. Synthetic worlds populate the utility, baseline, descriptor, and seed fields directly; benchmark-specific cost channels are set to zero when absent.

Field	Level	Purpose
Task identifier	task	Identifies task-level or rollout-level partitions
Candidate identifier	task-candidate-rollout	Supports deterministic tie-breaking and subset reconstruction
Seed and replicate index	run / rollout	Exact rerun of worlds and noise draws
Raw reward and cost components	rollout	Audit for clipped normalization and future benchmark loaders
Clipped utility $Y_{i,e,r}$	rollout	Primitive quantity entering all estimators
Cell mean $\hat{u}_{i,e}$	task-candidate	Matrix consumed by the selector
Baseline cell $\hat{u}_{i,0}$	task	Required for post-clipping subtraction
Task descriptor x_i	task	Required only for routing diagnostics
Selected subset S	run	Reproducible evaluation on hold-out matrices
Held-out oracle and routed values	run	Keeps selection and deployment diagnostics separate

Table 27: Selection complexity for an $n \times m$ matrix and budget k, excluding rollout acquisition.

Method	Complexity	Status
Reference greedy	$O(nmk^2)$	Used in archived analyses
Greedy with current-max cache	$O(nmk)$	Available optimization
Singleton ranking	$O(nm + m \log m)$	Used in archived analyses
Random subset and evaluation	$O(k + nk)$	Used in archived analyses
Exhaustive optimum	$O(nk\binom{m}{k})$	Used at stated budgets

Each Y is already clipped. Because clipping is nonlinear, the identity does not equate a difference of clipped utilities with a clipped difference of raw scores.

Synthetic worlds without token, API or tool-use costs log the corresponding fields as zero.

Table 26 lists the minimum fields saved for each rollout.

For example, with $m = 10$, $n = 128$, and $q = 12$, the literal calibration path stores $n(m + 1)q = 128 \times 11 \times 12 = 16{,}896$ rollout utilities. If held-out evaluation uses $n_{\text{test}} = 512$, it stores another $n_{\text{test}}(m + 1)q = 512 \times 11 \times 12 = 67{,}584$.

J.3 Greedy implementation, caching, and deterministic tie-breaking

The reference `greedy_select` recomputes the pointwise maximum over the selected columns for each candidate. At step t, this scans $t + 1$ columns per task and candidate, giving $O(nmk^2)$ work across the run. Exhaustive optimization similarly costs $O(nk\binom{m}{k})$. These are the implementations used for the archived analyses.

An exact optimized implementation can maintain $M_i^{(t)} = \max_{a \in S_t \cup \{0\}} \hat{u}_{i,a}$, score a candidate by $n^{-1} \sum_i [\max(M_i^{(t)}, \hat{u}_{i,e}) - M_i^{(t)}]$, and update the maxima after selection. This takes $O(nmk)$ work. The distinction concerns implementation cost, not the coverage objective or greedy guarantee.

Greedy scans a fixed candidate order and updates only for an improvement greater than 10^{-12}; ties therefore keep the earlier candidate. The historical singleton selector uses reversed NumPy `argsort` of singleton values. We retain these rules to reproduce the archived results; they are not a common smallest-index rule. Since the binary matrix contains many ties, sensitivity to ordering is a limitation of the comparison.

J.4 SYNTHETIC WORLD GENERATORS AND HYPERPARAMETERS

The synthetic studies generate utility matrices, calibration observations and, for routing, task descriptors. Their configurable components are the world generator, descriptor space and routing metric.

The latent-region generator used for the main held-out oracle sweeps samples a task region $g_i \in \{1, \dots, G\}$ and a skill region $h_e \in \{1, \dots, G\}$. The raw success template is

$$s_{i,e}^{\text{raw}} = 0.9\,\mathbf{1}\{g_i = h_e\} + 0.1\,\mathbf{1}\{g_i \neq h_e\} + \xi_{i,e}, \qquad \xi_{i,e} \sim \text{Unif}[-0.05,\, 0.05],$$

and the adjusted utility used by the selector is $u_{i,e} = \text{clip}\big(s_{i,e}^{\text{raw}} - c_e,\, 0,\, 1\big)$. The baseline raw success in that harness is $s_{i,0}^{\text{raw}} = 0.08$, the adjusted baseline is $u_{i,0} = \text{clip}(0.08 - 0.02, 0, 1) = 0.06$, and the candidate cost on the Stage 1 latent harness is sampled as $c_e \sim \text{Unif}[0.02,\, 0.15]$.

The redundant-clone construction for Theorem 1 fixes $m = k + (k-1)$. The free parameter γ sets the diverse-to-clone mass ratio. Given γ, the clone-region mass p and diverse-region mass q are

$$p = \frac{0.85}{1 + (k-1)\gamma}, \qquad q = \gamma p,$$

so that $q/p = \gamma$ and the total support mass is $p + (k-1)q = 0.85$. In this construction, p and q denote region masses, matching the notation of Theorem 1; this q is unrelated to the per-cell rollout count q of the calibration protocol.

The clean routing diagnostic uses descriptor space $x_i \in \mathbb{R}^d$ with Euclidean metric $d(x, x') = \|x - x'\|_2$. The nearest calibration descriptor for routing and the routed skill are

$$i(x) = \arg\min_{j \in \{1,\dots,n_{\text{cal}}\}} \|x - x_j\|_2, \qquad \rho_S(x) = \arg\max_{e \in S \cup \{0\}} \hat{u}_{i(x),e}.$$

For example, if $x = (0.20, 0.70)$ and the nearest calibration descriptor is $x_{i(x)} = (0.25, 0.65)$, the distance in Theorem 3 is $d(x, x_{i(x)}) = \sqrt{(0.20 - 0.25)^2 + (0.70 - 0.65)^2} \approx 0.071$.

Tables 28 and 29 collect the executed hyperparameters.

J.5 SPLIT DISCIPLINE AND REPRODUCIBILITY CONTROLS

The candidate pool is fixed before calibration; in the intended data protocol, this is $V \perp\!\!\!\perp \{Z_i^{\text{cal}}\}_{i=1}^n$. Synthetic suites use independently sampled calibration and test tasks. The primary MCPAgentBench and SkillsBench matrix comparisons retain the same task identities and separate rollout repetitions. SkillsBench uses complete measured observations; no observation is copied between partitions. These matrix comparisons support rollout-level stability on the evaluated tasks. The exploratory SkillsBench leave-one-task-out refits exclude each test task from fitting, but reuse previously examined data and heavily overlapping training sets. The separate 420-rollout transfer study freezes libraries before collecting outcomes on 14 new tasks; its category-balanced sampling and scoring amendment are documented in Appendix G.4.

Synthetic runs use explicit seeds and are indexed by $\omega = (\text{suite name}, \text{hyperparameter tuple}, \text{seed})$. The selected subset is stored as an ordered tuple $S_\omega = (e_1, \dots, e_k)$ to preserve tie order and marginal-gain traces.

The aggregation reported in Section 4 uses $N_{\text{seeds}} = 30$ for the main latent-world sweeps. An illustrative aggregation formula is $\overline{\text{Gap}} = \frac{1}{N_{\text{seeds}}} \sum_{s=1}^{N_{\text{seeds}}} \text{Gap}^{(s)}$, with standard error

$$\text{SE} = \frac{\sqrt{\frac{1}{N_{\text{seeds}} - 1} \sum_{s=1}^{N_{\text{seeds}}} \left(\text{Gap}^{(s)} - \overline{\text{Gap}}\right)^2}}{\sqrt{N_{\text{seeds}}}}.$$

Table 30 summarizes the minimum reproducibility checklist.

Table 28: Primary synthetic hyperparameters. The final column identifies the purpose and scope of each suite.

Suite	Main grid	Fixed defaults	Seed count	Status
Redundant clone	$k \in \{2, 3, 5\}$, $q/p \in \{0.95, 0.98\}$	Total support mass 0.85	Deterministic construction	Direct Theorem 1 support
Latent held-out sweeps	$n \in \{32, 64, 128, 256\}$, $q \in \{1, 2, 4, 8, 16\}$, $m \in \{6, 8, 10, 12\}$, $k \in \{1, 2, 3, 4, 5\}$	$G = 6$, default $(n, q, m, k) = (128, 12, 10, 4)$	30 in the main aggregation	Direct fixed-matrix support
Theorem 2 scaling	n- and q-sweeps	One synthetic path uses effective Gaussian cell noise	Stage 3 aggregation	Qualitative consistency only
Clean routing diagnostic	$L \in \{2, 4, 8\}$, $n_{\text{cal}} \in \{8, 32, 128\}$	$d = 2$, $\lvert S\rvert = 3, 8$ centers	Stage 3 aggregation	Conditional Theorem 3 support
Descriptor corruption	{`clean`, `random_embed`, `missing_var`, `aliased`}	$d = 3$, $\lvert S\rvert = 3$, $n_{\text{cal}} = 128$	Stage 3 aggregation	Out-of-regime falsification

Table 29: Settings for the perturbation, pool-construction, interaction and noise diagnostics.

Diagnostic	Hyperparameters	Role
Max-stability	Entrywise perturbation $\alpha \in \{0, 0.05, 0.10, 0.20\}$	Lemma-level audit of $\sup_S \lvert\tilde{F}(S) - F(S)\rvert \le 2\alpha$
Greedy transfer	Uniform perturbation magnitude matched to α-sweeps	Lemma-level audit of transfer under objective mismatch
Adaptive leakage	{`clean_split`, `leakage_pool_from_failures`}	Out-of-regime falsification of fixed-pool independence
Interaction stress	Synergy fraction $\gamma \in \{0, 0.1, 0.2, 0.4, 0.6\}$	Out-of-regime check for max-coverage misspecification
Heavy-tail noise	{`gauss`, `student_t`, `ar1`}	Out-of-regime noise robustness diagnostic

J.6 Compute accounting and benchmark feasibility

The dominant cost on a real benchmark is full-matrix calibration, not the greedy search itself. On the literal repeated-rollout path, the calibration, held-out, and total rollout budgets are

$$C_{\text{cal}} = (m+1)nq, \qquad C_{\text{test}} = (m+1)n_{\text{test}}q, \qquad C_{\text{total}} = (m+1)(n+n_{\text{test}})q.$$

Table 30: Reproducibility checklist for the matrix-based protocol.

Requirement	Implementation rule
Deterministic subset selection	Fixed ordering; greedy scan rule and archived singleton sorting rule documented above
Seeded world generation	Every synthetic matrix is generated from an explicit seed tuple
Split integrity	Independent synthetic tasks; disjoint rollout repetitions for fixed-task real studies
Selector audit trail	Save subset S, stepwise marginal gains, and objective values
Matrix audit trail	Save the full calibration and held-out matrices used for reporting
Routing audit trail	Save descriptors, nearest neighbors, routed skills, and routed regret
Figure reproducibility	Figures are generated only from saved summary arrays, never from transient in-memory state
Claim discipline	Mark literal-rollout and surrogate-noise runs separately in output metadata

Table 31: Live-evaluation entry points inspected for future studies. MCPAgentBench (Section 4.4) uses released scores, while SkillsBench (Section 4.1) uses a separate live-rollout harness. The listed entry points were not used to acquire the reported data.

Benchmark	Candidate entrypoint	Public assets observed	Status here
ALFWorld Shridhar et al. (2021)	`scripts/run_eval.py`	configs and environment data present	Reconnaissance only
MCPAgentBench	`runbenchmark.py`	configs and task JSON files present	Reconnaissance only (live acquisition; fixed-matrix study executed separately)
MCP-Bench Wang et al. (2025)	`run_benchmark.py`	config and task files present; some external API keys required	Reconnaissance only

At the default synthetic scale $(m, n, n_{\text{test}}, q) = (10, 128, 512, 12)$, a full-matrix study with repeated rollouts would require $C_{\text{total}} = 11 \times (128 + 512) \times 12 = 84{,}480$ rollout utilities.

Generating this many synthetic utility values is inexpensive; collecting them through agent rollouts is costly. The live SkillsBench study (Section 4.1) collected two repetitions across 30 tasks and four models for 16 candidate directories plus no-skill. The corrected 15-skill analysis retains 3,840 of those 4,080 observations. Acquisition cost is the main reason we did not run live studies on interactive benchmarks such as ALFWorld Shridhar et al. (2021). The MCPAgentBench study (Section 4.4) (MCPAgentBench contributors, 2025) uses third-party scores released in the benchmark repository; we collected no MCPAgentBench rollouts.

Selection reuses the collected matrix. The reference greedy implementation's selected-column scan count is $n \sum_{t=0}^{k-1}(m-t)(t+1) = 10{,}240$ at this scale; a cache reduces it to $n \sum_{t=0}^{k-1}(m-t) = 4{,}352$. For a fixed library and held-out matrix, oracle evaluation costs $C_{\text{oracle_eval}} = n_{\text{test}}(k+1)$; at the same scale, $C_{\text{oracle_eval}} = 512 \times 5 = 2{,}560$.

These counts describe array operations after matrix acquisition, not rollout, token or wall-clock costs. Appendix H.1 examines retrospective row subsampling.

Table 31 lists live-evaluation entry points inspected during feasibility work but not used in the reported studies.

K ARCHIVED SYNTHETIC RESULTS

This appendix collects the archived numerical summaries for the synthetic studies. The archive preserves seed-aggregated results and selected diagnostics, but not every individual seed result. The three seed-aggregation input files are byte-identical copies; they do not provide independent replication. Their stored means reproduce the reported values, while individual-seed variance cannot be re-audited from those copies. The tables identify quantities unavailable from these summaries. Real-data results appear in Appendices G and H.

The standard error used throughout the archived summaries is

$$\mathrm{SE} = \frac{\mathrm{sd}}{\sqrt{N_{\mathrm{seeds}}}}. \tag{5}$$

The normal-approximation confidence interval reported below is

$$\mathrm{CI}_{95} = \bar{x} \pm 1.96\,\mathrm{SE}. \tag{6}$$

The archived improvement metric used for baseline comparisons is

$$\mathrm{Advantage} = \mathrm{Gap}_{\mathrm{baseline}} - \mathrm{Gap}_{\mathrm{SC}}. \tag{7}$$

The claim-status field in the mapping table takes values in

$$\mathrm{Status} \in \{\text{proved}, \text{synthetic support}, \text{untested}\}\,. \tag{8}$$

Tables 32 and 33 record which claims the archived studies address and which remain outside their scope. The numerical summaries follow by suite.

For the redundant-clone construction, the pointwise-max objective predicts a limiting singleton ratio of

$$\frac{F(S_{\mathrm{singleton}})}{F(S^\star)} = \frac{1}{1 + (k-1)\,q/p}. \tag{9}$$

Table 36 lists the full theoretical grid used by the executed construction together with the archived empirical extrema.

Table 40 lists the inferential quantities available from the archived summaries. A paired p-value cannot be recovered when paired seed-wise differences are absent.

The additional diagnostics use the same selection objective under perturbations and changes to the assumptions. The stability diagnostic measures

$$\Delta_{\mathrm{stab}}(S) = \left|F_{\tilde{U}}(S) - F_U(S)\right|. \tag{10}$$

and the adaptive-pool diagnostic measures the optimism gap

$$\mathrm{Optimism} = F_{\mathrm{cal}}(S) - F_{\mathrm{test}}(S). \tag{11}$$

Tables 34–46 document fixed-pool synthetic coverage under the archive limits above. The studies do not resolve skill-use mechanisms, tight constants, paired significance for legacy summary-only contrasts, or deployment guarantees without descriptor assumptions.

Table 32: Formal claims and archived synthetic evidence. Proofs appear in Appendices B–D; numerical evidence here comes from the executed synthetic studies.

Claim	Reference	Empirical target	Archived result	Status
Coverage structure and greedy approximation for a fixed pool	Theorem 1	Redundant-clone construction and latent held-out sweeps	Greedy/OPT $= 1.000$ on clone worlds; worst singleton/OPT $= 0.2052$; mean held-out Skill-Cover gap $= 0.0106$	Proved + synthetic support
Noisy full-matrix transfer with separate n- and q-terms	Theorem 2	n-sweep and q-sweep scaling	Empirical slopes -0.40 in n and -0.50 in q; Spearman $0.71/0.83$	Proved + qualitative synthetic support
Conditional deployment under descriptor regularity	Theorem 3	Clean Lipschitz routing and descriptor corruption	Clean regret $= 0.013$; corrupted descriptors: 0.178, 0.135, 0.074; clean-regime regret/bound ratio 1% to 5%	Proved + synthetic support
Bounded normalization after clipping	Lemma 1	Implementation audit	Logged as an implementation requirement in Appendix J; no separate scalar summary	Proved
Pointwise-max diminishing returns	Lemma 2	Clone separation and coverage-style sweeps	Same synthetic evidence as Theorem 1; no archived violation-count table	Proved + indirect synthetic support
Finite-library uniform convergence	Lemma 4	n-sweep consistency	Spearman 0.71 for n-term qualitative fit	Proved + qualitative synthetic support
Max-stability under entrywise perturbation	Lemma 5	Entrywise perturbation audit	Archived plot summary reports empirical deviation at roughly 5% of the 2α bound	Proved + appendix-only synthetic support
Greedy transfer under uniform objective perturbation	Lemma 6	Perturbed-objective audit	Archived plot summary reports near-zero transfer gap for greedy-family selectors; top-k marginal and random remain around 0.12 to 0.17	Proved + appendix-only synthetic support
Interactive transfer beyond the evaluated fixed pools	Section 6	New pools with skill discovery or composition	Not evaluated here	Untested

Table 33: Claims not established by the reported studies.

Claim not established	Reason	Evidence status
SkillCover improves real deployed agents on ALFWorld	No ALFWorld fixed-matrix study was executed	Untested
SkillCover improves live MCP-style tool-use agents	The executed MCPAgentBench study is fixed-matrix selection on released scores, not live agent improvement	Fixed-matrix evidence only
The selection theorem implies end-to-end routed utility without extra assumptions	Theorem 3 is conditional and requires descriptor regularity; F, F_v, and J are separated in the theory	Not claimed
The method reduces rollout cost relative to full-matrix acquisition	The paper analyzes what to do after a full matrix exists; it does not solve matrix acquisition	Not claimed
The theory covers multi-skill composition, sequential planning, or interference	The pointwise-max model is one-skill-per-task and excludes interactions	Not claimed
Theorem 2 was validated with literal repeated-rollout q-averaging in every executed scaling fit	One Stage 3 path uses Gaussian surrogate cell noise scaled by $1/\sqrt{q}$	Not claimed
The main bounds are tight or practically sharp	Appendix-only audits show the constants are loose by one to two orders of magnitude	Not claimed

Table 34: Seed-aggregation stability for the main latent-world study. The table reports the archived summary across the sweep family used in Section 4. Confidence intervals are computed from equation 6.

N_{seeds}	Mean SC gap	95% CI (SC gap)	Mean SE	Adv. vs singleton	Adv. vs random
6	0.0072	[0.0005, 0.0139]	0.0034	0.1379	0.1201
12	0.0099	[0.0030, 0.0168]	0.0035	0.1291	0.1208
20	0.0108	[0.0049, 0.0167]	0.0030	0.1328	0.1214
30	0.0106	[0.0057, 0.0155]	0.0025	0.1340	0.1208

Table 35: Held-out latent-world results reported in Section 4. All rows use synthetic, fixed-pool, full-matrix evaluation.

Archived quantity	Value	Seeds	In archive	Note
Overall mean SkillCover held-out oracle gap	0.0106	30	Yes	Overall latent-world summary
Overall mean advantage over singleton	0.1340	30	Yes	Defined by equation 7
Overall mean advantage over random	0.1208	30	Yes	Defined by equation 7
$k = 4$ SkillCover held-out oracle gap	0.0108	30	Yes	Explicitly cited in Section 4
$k = 4$ singleton held-out oracle gap	0.1943	30	Yes	Explicitly cited in Section 4
$k = 4$ absolute gap difference: singleton minus SkillCover	0.1835	30	Yes	Computed from archived means
Worst redundant-clone singleton/OPT ratio	0.2052	determ. constr.	Yes	Reported extremal row
Best redundant-clone greedy/OPT ratio	1.0000	determ. constr.	Yes	Reported across clone rows

Table 36: Redundant-clone grid. The construction gives the ratio in equation 9. The available summaries preserve the grid specification and extremal empirical outcomes, but not every individual Monte Carlo row.

k	q/p	Singleton/OPT limit	Greedy/OPT	Archived empirical note
2	0.95	0.5128	1.0000	Individual singleton row not preserved in summaries
2	0.98	0.5051	1.0000	Individual singleton row not preserved in summaries
3	0.95	0.3448	1.0000	Individual singleton row not preserved in summaries
3	0.98	0.3378	1.0000	Individual singleton row not preserved in summaries
5	0.95	0.2083	1.0000	Individual singleton row not preserved in summaries
5	0.98	0.2033	1.0000	Worst archived singleton/OPT row reported as 0.2052
Archived extrema over all clone rows		—	1.0000	Singleton/OPT minimum $= 0.2052$

Table 37: Qualitative scaling summary for Theorem 2, as reported in Section 4. The q-path is closer to the ideal $-1/2$ slope than the n-path. One path uses surrogate Gaussian cell noise, so these fits do not validate every repeated-rollout assumption.

Panel	Predicted trend	Slope	Rank corr.	Interpretation
n-sweep	$\log(\text{regret}) = a_n + b_n \log n$, $b_n \approx -0.5$	-0.40	0.71	Qualitatively consistent, slightly shallower than ideal rate
q-sweep	$\log(\text{regret}) = a_q + b_q \log q$, $b_q \approx -0.5$	-0.50	0.83	Very close to the nominal $q^{-1/2}$ trend

Table 38: Archived descriptor-based routing diagnostics. The clean row is the reference; the other rows use corrupted descriptors.

Descriptor mode	Regret	Δ vs clean	$\times$ vs clean	Role in paper
Clean	0.013	0.000	1.00×	In-regime synthetic support
Random embedding	0.178	0.165	13.69×	Out-of-regime falsification
Missing variable	0.135	0.122	10.38×	Out-of-regime falsification
Aliased descriptor	0.074	0.061	5.69×	Out-of-regime falsification

Table 39: Clean-Lipschitz routing results from the archived reconciliation note. The range of regret-to-bound ratios is preserved; individual Lipschitz and calibration-size rows are unavailable.

Archived quantity	Value	In archive	Note
Clean-descriptor mean routed regret	0.013	Yes	Same quantity as Table 38
Empirical regret / theorem-bound ratio	0.01 to 0.05	Yes	Reported range over clean Lipschitz worlds
Calibration density sweep	$n_{\text{cal}} \in \{8, 32, 128\}$	Yes	Executed grid described in Appendix J
Lipschitz sweep	$L \in \{2, 4, 8\}$	Yes	Executed grid described in Appendix J

Table 40: Intervals and tests available for the archived comparisons. Paired tests require seed-wise differences, which are absent from the available summaries.

Comparison	Archived point estimate	Archived interval / test	Identifiable from archive
Overall SkillCover mean gap at $N_{\text{seeds}} = 30$	0.0106	95% CI $[0.0057, 0.0155]$	Yes
Overall SkillCover mean gap at $N_{\text{seeds}} = 20$	0.0108	95% CI $[0.0049, 0.0167]$	Yes
Overall advantage over singleton at $N_{\text{seeds}} = 30$	0.1340	Exact paired p-value unavailable	No
Overall advantage over random at $N_{\text{seeds}} = 30$	0.1208	Exact paired p-value unavailable	No
$k = 4$ singleton minus SkillCover gap difference	0.1835	Exact paired p-value unavailable	No
Random-embedding minus clean routing regret	0.165	Exact paired p-value unavailable	No
Missing-variable minus clean routing regret	0.122	Exact paired p-value unavailable	No
Aliased minus clean routing regret	0.061	Exact paired p-value unavailable	No

Table 41: Availability of individual seed results in the summary archive. Aggregate results are retained to the extent stated in each row; individual seed records are not printed here.

Suite	Seeds	Mean/SE in archive	Seed rows printed	Comment
Latent held-out sweeps	$6, 12, 20, 30$	Yes	No	Means and SEs are preserved; individual rows are absent from these summaries
Redundant-clone suite	deterministic grid	Extremal ratios preserved	No	Full grid specification preserved; all singleton rows not printed
Theorem 2 scaling fits	Stage 3 aggregation	Yes	No	Slopes and Spearman values preserved
Clean-Lipschitz routing	Stage 3 aggregation	Partial	No	Mean regret and bound-tightness range preserved
Descriptor corruption	Stage 3 aggregation	Yes	No	Four mean regrets preserved
Adaptive leakage appendix audit	Stage 3 aggregation	Yes	No	Mean optimism rows preserved

Table 42: Archived perturbation and transfer diagnostics for Lemmas 5 and 6. Numerical summaries are preserved, but full grids are unavailable.

Audit	Archived numerical summary	Selector dependence	Interpretation
Max-stability perturbation	Empirical deviation is about 5% of the 2α bound	Essentially algorithm-agnostic in archived summary	Supports Lemma 5; constants are loose
Greedy-transfer perturbation	Greedy-family transfer gap is near 0; top-k marginal and random remain around 0.12 to 0.17	Strong selector dependence	Supports Lemma 6; poor selectors do not inherit transfer

Table 43: Optimism under fixed and adaptive pools. Adaptive pools depend on calibration failures, outside the fixed-pool theorem assumptions.

Selector	Clean optimism	Leakage optimism	Δ	Leakage/clean
Greedy	0.007	0.029	0.022	4.14×
Random	0.016	0.014	−0.002	0.88×
Top-k marginal	0.026	0.029	0.003	1.12×
Stochastic lazy greedy	0.010	0.018	0.008	1.80×
Local search	0.020	0.045	0.025	2.25×

Table 44: Archived plot summaries for diagnostics that violate theorem assumptions or compare auxiliary selectors.

Diagnostic	Archived numerical range	Role
Submodularity stress with AND-gate synergy	Greedy-family $F(S)/\mathrm{OPT} \approx 1.0$ over $\gamma \in [0, 0.6]$; random drops to about 0.43 at high γ	Out-of-regime robustness check
Heavy-tail / AR(1) noise	Greedy-family held-out gap about 0.007 to 0.03; top-k marginal reaches about 0.21 under AR(1)	Out-of-regime robustness check
Descriptor corruption	Routing regret inflates by about 5× to 14× relative to clean descriptors	Assumption-falsification check for Theorem 3

Table 45: Study status. Real-data evidence includes released MCPAgentBench scores and our SkillsBench experiments. ALFWorld and live MCP-Bench acquisition remain untested.

Benchmark or study	Intended role	Executed here	Reported numerical rows
Synthetic redundant-clone suite	Support Theorem 1	Yes	Yes
Synthetic latent held-out sweeps	Headline fixed-pool evidence	Yes	Yes
Synthetic n- and q-scaling fits	Qualitative support for Theorem 2	Yes	Yes
Synthetic clean-Lipschitz routing	Conditional support for Theorem 3	Yes	Partial summary only
Synthetic descriptor corruption	Boundary-case diagnostic	Yes	Yes
MCPAgentBench fixed-matrix study	Real benchmark	Yes	Section 4.4
SkillsBench calibration and historical mounting	Real-skill matrix and deployment	Yes	Section 4.1
SkillsBench new-task transfer	Frozen-library deployment	Yes	Appendix G.4
SkillsBench fresh-calibration diagnostic	Calibration and deployment variability	Yes	Appendix G.5
ALFWorld live-acquisition study	Real benchmark (interactive)	No	None
MCP-Bench live-acquisition fallback	Real benchmark (interactive)	No	None

Table 46: Key archived synthetic results used in Sections 3 and 4.

Quantity	Value	Interpretation
Mean latent-world SkillCover gap	0.0106	Mean held-out oracle coverage error at 30 seeds
Mean advantage over singleton	0.1340	Greedy advantage over singleton in the latent-world study
Worst clone-world singleton/OPT	0.2052	Near-$1/k$ singleton ratio in the clone construction
Clone-world greedy/OPT	1.0000	Greedy attains the optimum in the clone construction
n-sweep slope	-0.40	Qualitative consistency with the task-sampling term in Theorem 2
q-sweep slope	-0.50	Qualitative consistency with the rollout-noise term in Theorem 2
Clean routing regret	0.013	Low reported regret with clean descriptors
Random-embed routing regret	0.178	Higher reported regret under random descriptors