SKILLALIGN: TASK-AWARE SKILL EVALUATION THROUGH CAPABILITY–QUALITY ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Agent skills provide a lightweight mechanism for improving agent behavior through reusable procedural knowledge, without modifying model parameters. Yet as skill libraries grow, a fundamental question remains underexplored: what makes a skill useful for a particular task? Existing evaluation typically treats skill quality as either an intrinsic property of the skill or an empirical outcome measured through downstream execution. The former overlooks heterogeneous task requirements, while the latter is costly and offers limited guidance for skill selection and improvement before execution. We argue that skill utility should instead be characterized by the alignment between what a task requires and what a skill provides. We introduce SkillAlign, a task-aware framework for modeling this alignment prior to execution. SkillAlign represents tasks through six functional capability requirements and skills through six structured quality dimensions, connected by a capability-to-quality mapping. It computes capability-level task–skill matches using task-specific importance and primary-axis gating, preventing secondary qualities from compensating for deficiencies in capability-critical dimensions. The resulting alignment score supports skill ranking and selection, while capability-level deficiencies provide diagnostic signals for skill optimization. We evaluate SkillAlign across WebShop, ALFWorld, SWE-bench, and WebArena using complementary evidence from task profiling, controlled skill ranking, pairwise selection, quality-guided optimization, cross-model downstream execution, and mapping robustness. The results show descriptive variation among the assessed task profiles, task-conditioned evaluation discriminates high-quality Skills associated with successful executions from low-quality Skills associated with failed executions more effectively than holistic judgment or task-agnostic quality aggregation, and optimized skills consistently improve downstream performance across the tested agent models and benchmarks. Together, these findings establish task–skill alignment as a practical basis for pre-execution skill evaluation, selection, diagnosis, and optimization.

1 INTRODUCTION

Large language model (LLM) agents increasingly rely on reusable skills: structured procedural knowledge that specifies how to search, plan, use tools, preserve constraints, verify outcomes, or recover from failure. Skills can be acquired from trajectories, generated automatically, or maintained in shared libraries (Wang et al., 2023; Liang et al., 2026; Ni et al., 2026; Yang et al., 2026b; Huang et al., 2026). As skills become independently created, accumulated, and reused across tasks, a fundamental question arises: what makes a skill useful for a particular task Answering this question is important not only for selecting among candidate skills, but also for diagnosing their limitations and improving them before costly downstream execution.

Skill utility cannot be reduced to an absolute notion of quality. A skill that provides strong search and planning support may be highly useful for a task dominated by sequential decision making, yet offer limited benefit when the main bottleneck lies in verification or recovery. Conversely, a skill may be coherent, detailed, and generally well constructed while failing to address the capabilities most critical to the target task. We therefore distinguish *intrinsic skill quality*—the properties of the skill itself—from *task-specific utility*—the extent to which those properties support the functional

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.

requirements of a particular task. From this perspective, evaluating a skill requires modeling not only what the skill provides, but also what the task demands.

Existing evaluation paradigms capture only parts of this relationship. Agent benchmarks primarily measure utility through interactive outcomes (Yao et al., 2022a; Shridhar et al., 2020; Jimenez et al., 2023; Zhou et al., 2023; Liu et al., 2023a), providing direct evidence of effectiveness but requiring downstream execution. Model-based evaluators offer scalable assessment of generated artifacts (Liu et al., 2023b; Zheng et al., 2023; Zhu et al., 2023), while recent work has begun to benchmark the utility, generation, or security of agent skills (Li et al., 2026b; Han et al., 2026; Wang et al., 2026; Yu et al., 2026). These directions provide essential evidence, but leave a complementary question insufficiently addressed: how can the potential utility of a skill be characterized in relation to the requirements of a target task before execution? A task-independent quality score does not explicitly represent this compatibility, while execution outcomes reveal whether a skill worked without necessarily identifying which task-relevant properties made it suitable or deficient.

We introduce SkillAlign, a task-aware framework that operationalizes skill utility through *task–skill alignment*. The framework represents a target task by six functional capability requirements and independently evaluates a skill along six structured quality dimensions. A frozen capability-to-quality mapping specifies how quality dimensions support different functional capabilities. For each capability, SkillAlign combines its task-specific importance with the corresponding skill support through a primary-axis gate, which preserves the dominant role of the capability-critical quality dimension while allowing bounded contributions from secondary dimensions. The resulting capability-level matches are aggregated into an alignment score for pre-execution ranking and selection. The same representation exposes task-relevant deficiencies that can be used to guide targeted skill optimization.

This formulation leads to a unified evaluation perspective. Rather than asking only whether a skill is "good" in isolation, SkillAlign asks whether its strengths correspond to the capabilities that matter for the target task. This distinction makes it possible to use the same structured representation across evaluation, selection, diagnosis, and optimization, while keeping downstream execution as an independent source of empirical validation.

Our contributions can be summarized as follow:

1. We formulate agent-skill evaluation as a task–skill alignment problem, explicitly separating intrinsic skill quality from task-specific utility.
2. We introduce SkillAlign, which connects six task-capability requirements with six skill-quality dimensions through a frozen capability-to-quality mapping. The resulting alignment supports pre-execution ranking and selection, while capability-level deficiencies provide diagnostic signals for targeted skill optimization.
3. We evaluate SkillAlignacross WebShop, ALFWorld, SWE-bench Lite, and WebArena with two agent backbones. Experiments examine predictive ranking and pairwise selection, task-aware optimization and downstream utility, and robustness to mapping, evaluator, and hyperparameter choices.

2 Related Work

LLM-based evaluation. LLM-based evaluators increasingly use structured criteria and explicit rubrics to improve judgment reliability. G-Eval introduces chain-of-thought-based form filling for natural-language generation evaluation, while MT-Bench and JudgeLM study and improve LLM-as-a-judge evaluation at scale (Liu et al., 2023b; Zheng et al., 2023; Zhu et al., 2023). Our setting differs in that the object of evaluation is a reusable Skill whose utility may vary across target tasks. Thus, beyond assessing intrinsic quality along explicit dimensions, we condition their relevance on task-specific capability requirements.

Agent Skill evaluation and optimization. Recent work has explored the acquisition, reuse, transfer, and continual evolution of agent Skills (Liang et al., 2026; Li et al., 2026b; Han et al., 2026; Ni et al., 2026; Yang et al., 2026b). Other studies examine Skill quality, utility, and reliability through dedicated evaluation or testing procedures (Gao et al., 2026; Ding et al., 2026; Wang et al., 2026; Xu & Yan, 2026). These efforts establish the importance of evaluating reusable Skills, but generally treat

Table 1: Task-capability and Skill-quality dimensions.

ID	Dimension	Definition / assessment focus
Task capabilities		
C_1	Goal Modeling	Identify objectives, constraints, and conditions for successful completion.
C_2	State Modeling	Maintain task-relevant state and track changes induced by interaction.
C_3	Planning	Construct intermediate decisions and actions toward the target state.
C_4	Execution	Translate intended decisions into valid environment or tool actions.
C_5	Verification	Assess intermediate and final outcomes and detect incorrect or incomplete progress.
C_6	Adaptation	Revise states, plans, or actions in response to feedback, failures, or environmental changes.
Skill qualities		
Q_1	Goal & Constraint Fidelity	Preserve objectives, constraints, and success criteria.
Q_2	Decision Policy Explicitness	Specify actionable, state-dependent decision rules.
Q_3	Search & Resource Convergence	Constrain search through priorities, budgets, and stopping rules.
Q_4	Evidence–State–Action Coupling	Ground actions in observations and maintained state.
Q_5	Verification & Recovery Completeness	Specify checks, failure detection, recovery, and closure.
Q_6	Intervention Cost & Compositional Calibration	Balance guidance benefit, intervention cost, and compatibility.

Skill quality or utility as the primary object of assessment. In contrast, we explicitly separate task capability requirements from intrinsic Skill quality and evaluate their compatibility through a frozen capability-to-quality mapping.

Related prompt and program optimization methods show that structured feedback can guide iterative improvement (Khattab et al., 2023; Pryzant et al., 2023; Fernando et al., 2023). We similarly use evaluation signals for optimization, but derive them from task-conditioned capability gaps, connecting the same representation to pre-execution Skill selection and targeted revision.

3 Methodology

3.1 Problem Formulation

Let T denote a target task and S an agent skill. Rather than treating utility as an intrinsic score $Q(S)$, we model it through a task-capability profile $\mathbf{C}_T$, a Skill-quality profile $\mathbf{Q}_S$, and a capability-to-quality mapping $\mathbf{M}$. The central hypothesis is that a Skill is effective when its quality characteristics support the capabilities required by the target task. The resulting alignment score $F(T, S)$ estimates this task-conditioned utility before execution.

3.2 Task Capability Modeling

Rather than defining capabilities around specific benchmarks, we derive a task-independent functional abstraction from recurring mechanisms in representative agent architectures. Seminal agent frameworks instantiate mechanisms for maintaining goals and contextual states, planning, action execution, outcome verification, and feedback-driven adaptation Yao et al. (2022b); Park et al. (2023); Shinn et al. (2023); Wang et al. (2023). Recent studies and surveys continue to characterize agent systems through closely related functional components, including reasoning and planning, memory and state maintenance, tool/environment interaction, self-verification, reflection, and adaptive control Yehudai et al. (2026); Zhou et al. (2024); Yang et al. (2026a); Xu (2026). We consolidate these recurring mechanisms into six functional dimensions—Goal Modeling, State Modeling, Planning, Execution, Verification, and Adaptation—which capture the major functional stages of agent–environment interaction. These dimensions are also natural intervention points for skills: skills can refine goals and constraints, maintain task-relevant state, guide planning and action selection, prescribe execution procedures, or introduce verification and recovery strategies. The resulting capability space therefore provides a task-independent interface for characterizing both task requirements and where skill intervention is needed.

Given this task-independent capability space, each task is represented by $\mathbf{C}_T = [c_{T,1}, \ldots, c_{T,6}]$, where $c_{T,i} \in \{0, \ldots, 5\}$ denotes the assessed opportunity for a short, fixed Skill intervention at capability C_i to improve task execution. Thus, $\mathbf{C}_T$ characterizes task-side opportunities for Skill support rather than the capability of a particular agent or the intrinsic difficulty of the task. The profile is estimated solely from task-side information—including the task specification, action space,

environment constraints, and success criteria—without access to candidate Skills or their execution outcomes.

3.3 Skill Quality Modeling

While the capability model characterizes *where* skill support is required in the agent process, the quality model characterizes *how well* a skill provides such support. We assess skill quality along six dimensions that capture complementary properties of effective intervention: preserving task intent, specifying decision logic, constraining search and resource use, grounding actions in evidence and state, supporting verification and recovery, and controlling the cost and composability of intervention. Together, these dimensions cover both the effectiveness and operational overhead of a skill, while remaining independent of its observed downstream outcome.

For a Skill S, the structured evaluator produces $\mathbf{Q}_S = [q_{S,1}, \ldots, q_{S,6}]$, where $q_{S,j} \in \{0, \ldots, 5\}$ measures its quality along dimension Q_j, from absent or counterproductive (0) to exceptional (5). The six dimensions and their assessment criteria are summarized in Table 1.

Each score must be justified by evidence in the skill specification rather than inferred from downstream performance. The evaluator receives the skill, the scoring rubric, and, when necessary, task context for interpreting task-specific goals or constraints. It has no access to execution outcomes, success/failure labels, empirical rankings, or competing skills. This separation prevents downstream effectiveness from leaking into the intrinsic quality assessment. Detailed scoring anchors and evaluator protocols are provided in Appendix A.

3.4 Capability-to-Quality Mapping

Task capabilities and skill qualities occupy different representation spaces. We connect them with $\mathbf{M} \in \mathbb{R}^{6\times 6}$, where M_{ij} is the relative contribution of quality dimension Q_j to capability C_i. Each row is nonnegative and sums to one. The mapping used in our experiments is provided in Appendix B.

For each capability C_k, its primary quality dimension is $p(k) = \arg\max_j M_{k,j}$.

The mapping is constructed independently of downstream outcomes. Evaluators first score the relevance of every $C_i \to Q_j$ relation using the dimension definitions, then normalize each row and aggregate evaluator matrices. The experimental protocol freezes the resulting matrix before candidate evaluation. Independent mapping assessment and weight perturbations test whether conclusions depend on a single subjective matrix (Section 4.6).

3.5 Task–Skill Alignment

A simple weighted sum allows high scores on secondary quality dimensions to compensate for a deficiency in the quality dimension most critical to a capability. To prevent such compensation, SkillAlign computes task–skill alignment at the capability level using a primary-axis gate.

For each capability C_k, we convert its task-side leverage $c_{T,k}$ into an importance coefficient:

$$a_k(T) = \left(\frac{c_{T,k}}{5}\right)^{\eta}. \tag{1}$$

Let $d_k = \sum_{j\neq p(k)} M_{k,j}$ denote the secondary mapping mass. The secondary support is the mapping-weighted average $u_k(S) = \sum_{j\neq p(k)} M_{k,j} q_{S,j}/d_k$. For the reported mapping, $d_k \geq 0.55$ for every row; the numerical safeguard for a vanishing denominator is specified in Appendix B.

The effective skill support for capability C_k is then

$$c_k(S) = q_{S,p(k)} \left[\alpha + (1-\alpha)\frac{u_k(S)}{5}\right]. \tag{2}$$

This primary-axis gate makes $q_{S,p(k)}$ the capacity-limiting factor: secondary dimensions can modulate its contribution only through a factor in $[\alpha, 1]$, rather than compensate for a weak primary dimension.

Define the capability-level match as $z_k(T, S) = a_k(T)c_k(S)$. We aggregate the two strongest capability-level matches so that alignment reflects the Skill's most task-relevant support rather than

being diluted by weakly relevant capabilities. Let $z_{(1)} \geq z_{(2)} \geq \cdots \geq z_{(6)}$ denote the sorted matches. The final alignment score is

$$F(T, S) = 20 \left[\beta z_{(1)} + (1 - \beta) z_{(2)}\right] \tag{3}$$

We use $\alpha = 0.7$, $\beta = 0.8$, and $\eta = 1$ as the representative configuration. Since $z_k(T, S) \in [0, 5]$, the resulting score satisfies $F(T, S) \in [0, 100]$.

For comparison, the task-agnostic baseline ignores capability requirements and mapping structure:

$$F_{\text{raw}}(S) = \frac{1}{6} \sum_{j=1}^{6} q_{S,j}. \tag{4}$$

Unlike this uniform aggregation, SkillAlign conditions skill quality on task-specific capability importance, preserves the dominant role of each capability's primary quality dimension, and aggregates the strongest capability-level matches.

3.6 Quality-Guided Skill Optimization

The capability-level formulation also provides a diagnostic signal for skill optimization. For each capability C_k, we define its task-relevant support gap as

$$G_k(T, S) = a_k(T) \left[5 - c_k(S)\right]. \tag{5}$$

A large $G_k(T, S)$ indicates that the capability is important to the target task but insufficiently supported by the current skill. Because $c_k(S)$ incorporates the primary-axis gate, the diagnostic naturally prioritizes deficiencies in the primary quality dimension while retaining the contribution of secondary dimensions.

The optimizer receives the original skill, capability and quality definitions, evidence-linked assessments, and the highest-priority capability gaps. For each prioritized gap $G_k(T, S)$, it revises the skill by primarily strengthening the associated primary quality dimension $Q_{p(k)}$, while also addressing the secondary dimensions linked to C_k by the mapping weights M_{kj}. The revision preserves the skill's original scope and avoids unnecessary intervention cost. The optimized skill is subsequently reassessed without access to downstream performance. Appendix C provides the revision protocol and stopping conditions.

4 Experiments

4.1 Experimental Setup

Benchmarks and skills. We evaluate on four agent benchmarks: WebShop (Yao et al., 2022a) for interactive shopping, ALFWorld (Shridhar et al., 2020) for embodied household tasks, SWE-bench Lite (Jimenez et al., 2023) for repository-level software repair, and WebArena (Zhou et al., 2023) for realistic web interaction. The source-skill pool is assembled from publicly accessible online skill repositories and benchmark releases, including SkillsMP (SkillsMP, 2026), SkillNet (Liang et al., 2026), SWE-Skills-Bench (Han et al., 2026), and SGDR (Li et al., 2026a). Not every evaluated text is an independently released artifact: the WebShop and ALFWorld S_1–S_6 candidates are controlled variants derived for the ranking study, and the SWE-bench LQ texts are length-controlled experimental derivatives of source Skills.

Models and conditions. Downstream experiments use GPT-5.4 (OpenAI, 2026) and GLM-5.3-Flash (Z.ai, 2026) at temperature 0, with four independent runs per configuration. DeepSeek-V4-Flash serves as the shared evaluator for task profiling, Skill-quality assessment, mapping assessment, and model-based baselines. We consider no-skill, origin, and optimized Skills; WebShop and ALFWorld additionally use six controlled quality variants $S_1, \ldots, S_6$, while RQ3 uses 30 paired high-quality and low-quality SWE-bench Skills selected from the 300-instance SWE-bench Lite evaluation set. Full configurations and selection criteria are provided in Appendix J and Appendix E.

Baselines and evaluation. We compare SkillAlign with three pre-execution baselines: Direct LLM, which makes a holistic pairwise judgment; Validated Rubric, which applies a published structured evaluation rubric (Huang et al., 2026); and Raw Q Mean, which uniformly averages the same six Skill-quality scores used by SkillAlign without task conditioning (Eq. 4). Predictive ranking is evaluated using Spearman's ρ and Kendall's τ, pairwise selection using accuracy, and downstream execution using each benchmark's primary performance metric. Voting procedures, confidence intervals, statistical tests, and benchmark-specific endpoints are detailed in Appendix J.

4.2 RQ1: Descriptive Task Capability Profiles

Figure 1 provides a descriptive comparison of the capability-requirement profiles assigned under our assessment protocol. The profiles differ in their emphasis across the six dimensions. WebShop places strong emphasis on planning, consistent with navigating product options toward a specified purchasing goal. ALFWorld emphasizes goal modeling and adaptation, consistent with maintaining objectives while responding to changes during embodied interaction. SWE-bench Lite places strong emphasis on planning and verification, reflecting the coordinated modifications and correctness checks involved in repository-level problem solving. WebArena exhibits a comparatively balanced profile, with relatively stronger goal modeling. These interpretations contextualize the assessed profiles rather than constitute independent evidence for the ratings.



Figure 1: Descriptive capability profiles of the four benchmarks across the six dimensions. The profiles represent assessed requirements without implying inferential significance.

The observed differences motivate examining whether Skill quality should be interpreted relative to task-specific capability requirements. Uniform quality aggregation does not represent such differences and may overlook whether a Skill supports the capabilities emphasized by a target task. This is a hypothesis motivated by the profiles, rather than a conclusion established by their visual separation. RQ1 characterizes the assessed benchmark profiles descriptively and does not claim statistically significant differences between benchmark populations. The predictive value of task-conditioned evaluation is tested separately in RQ2 and RQ3.

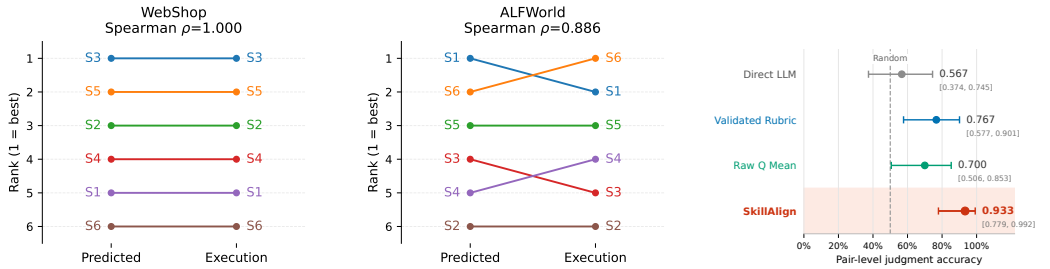
4.3 RQ2: Predictive Validity of Task-Aware Skill Evaluation

Figures 2a and 2b compare the Skill rankings predicted by SkillAlign with the empirical rankings obtained from downstream execution. Each experiment contains six deliberately controlled variants, one targeting each predefined quality dimension. The candidate set therefore covers the six intervention axes of the framework, but it should not be interpreted as a statistical sample from a broader population of Skills.

On WebShop, the predicted and empirical rankings are identical. On ALFWorld, 13 of the 15 pairwise orderings are recovered correctly. The two disagreements are the adjacent pairs S_1/S_6 at ranks 1–2 and S_3/S_4 at ranks 4–5. This ordering yields Spearman's $\rho = 0.886$ and Kendall's $\tau = 0.733$. We report these coefficients as compact descriptions of agreement within the six-variant controlled design, rather than as precise estimates of population-level ranking performance. Per-variant execution results for both agent backbones are reported in Tables 10 and 11.

4.4 RQ3: Is Task-Aware Evaluation Necessary?

Protocol. We construct 30 SWE-bench pairs, each containing a high-quality Skill (HQ) associated with a successful execution and a length-controlled low-quality Skill (LQ) associated with a failed execution. The control is deliberately conservative with respect to a possible HQ-length advantage: in 25 of 30 pairs the LQ text is longer than the HQ text (by 29.7–153.8%), while in the remaining five



(a) WebShop skill rankings. (b) ALFWorld skill rankings. (c) SWE-bench judgment accuracy.

Figure 2: Pre-execution predictions and observed outcomes. Panels (a) and (b) compare predicted and empirical Skill rankings on WebShop and ALFWorld, respectively. Panel (c) reports pairwise selection accuracy with 95% confidence intervals for Direct LLM, Validated Rubric, Raw Q Mean, and SkillAlign on SWE-bench.

pairs the HQ text is only 3.9–5.2% longer. We also rerun the failed Skills after length expansion; both the original and expanded versions fail all 30 paired tasks (0/30 successes), as reported in Table 8. We compare Direct LLM, Validated Rubric, Raw Q Mean, and SkillAlign using the protocol in Section 4.1. Accuracy is reported with 95% Clopper–Pearson intervals; comparisons use McNemar tests with Holm correction. Pair construction and vote-level analyses are provided in the appendix.

Results. As shown in Figure 2c, SkillAlign selects the HQ Skill in 28 of 30 pairs (93.3%; 95% CI: 77.9–99.2%), compared with 17/30 for Direct LLM (56.7%), 23/30 for Validated Rubric (76.7%), and 21/30 for Raw Q Mean (70.0%). The improvements over Direct LLM and Raw Q Mean are significant after Holm correction (adjusted $p = 0.0029$ and $p = 0.0313$); the improvement over Validated Rubric is numerical but not significant ($p = 0.0625$). Raw Q Mean and SkillAlign use identical quality scores, so their difference supports task-conditioned matching over uniform averaging. This comparison evaluates the complete matching and aggregation mechanism rather than individual components. The evidence is limited to 30 selected pairs, and the nonsignificant comparison with Validated Rubric warrants caution in interpreting that advantage.

4.5 RQ4: Does Task-Aware Optimization Improve Skill Utility?

Downstream utility. Table 2 compares No Skill, Origin Skill, General Optimize, Skill-Creator, and SkillAlign-guided optimization. General Optimize directly prompts an LLM to improve the origin Skill without structured feedback, whereas Skill-Creator follows Anthropic's publicly released procedure (Anthropic, 2026). Across both agent backbones, SkillAlign achieves the highest mean primary performance in every model–benchmark combination and the lowest mean step count on WebShop and ALFWorld. It consistently improves over Origin Skill, whereas General Optimize does not reliably improve performance and Skill-Creator remains below SkillAlign on all primary metrics.

Task-relevant quality changes. Figure 3 shows heterogeneous changes across quality dimensions and benchmarks, consistent with the gap-based objective: revisions depend jointly on task requirements and deficiencies in the origin Skill rather than uniformly increasing all scores. For example, WebShop shows pronounced gains in Search and Resource Convergence and Decision Policy Explicitness, while SWE-bench Lite gains substantially in Search and Resource Convergence and Verification and Recovery Completeness. Mean task–skill alignment increases by 12.88–18.61 points across the four benchmarks, further indicating task-dependent adjustments. Because the same framework guides optimization and reassesses Skill quality, these profile and alignment changes are diagnostic rather than independent effectiveness evidence; downstream execution provides the external validation. Full benchmark-level results are provided in Appendix C.

External optimization validation. We further apply SkillAlign-guided optimization, without access to downstream outcomes, to all 49 public Skills from SWE-Skills-Bench and rerun their paired tasks. As shown in Table 3, optimization improves mean performance from 92.3 to 94.2 with GPT-5.4 (+1.9 points) and from 72.8 to 77.3 with GLM-5.3-Flash (+4.5 points). The gains are also

Table 2: Downstream execution results, reported as mean ± sample standard deviation over four separate API runs with identical configurations and temperature set to 0. Best mean results within each model are shown in bold.

Method	WebShop			ALFWorld		SWE-bench Lite	WebArena
	Reward ↑	Acc.(%) ↑	Steps ↓	Acc.(%) ↑	Steps ↓	Resolved(%) ↑	Success(%) ↑
GPT-5.4							
No Skill	63.8±0.2	33.6±0.1	8.2±1.6	88.6±0.6	14.8±0.6	68.7±1.5	66.2±0.8
Origin Skill	63.6±0.4	34.2±0.3	7.6±2.2	93.4±1.4	13.1±0.8	70.0±0.7	67.8±0.5
General Optimize	62.8±0.6	33.2±0.7	7.9±2.1	88.3±1.2	14.6±1.5	67.7±0.9	65.5±0.4
Skill-Creator	64.7±0.2	33.7±0.3	8.2±2.5	93.1±1.4	13.3±1.7	72.3±0.3	67.9±0.2
SkillAlign (Ours)	**68.5±0.5**	**34.8±0.1**	**7.6±1.7**	**93.8±0.7**	**9.6±2.2**	**77.0±0.7**	**68.4±0.3**
GLM-5.3-Flash							
No Skill	58.2±0.7	29.2±0.4	12.3±2.3	84.3±0.4	16.9±1.5	46.7±0.5	45.3±0.4
Origin Skill	58.7±0.2	30.6±0.6	11.1±1.9	87.6±0.6	14.3±2.1	50.0±1.5	46.7±0.3
General Optimize	58.3±0.4	29.7±0.5	13.2±1.9	83.2±1.2	16.7±1.8	44.1±0.9	44.9±0.6
Skill-Creator	61.1±0.5	30.9±0.1	11.7±2.3	87.9±0.7	15.2±1.4	52.5±0.5	47.2±0.9
SkillAlign (Ours)	**62.4±0.4**	**31.4±0.2**	**10.9±2.6**	**88.9±0.4**	**12.9±1.6**	**55.3±0.7**	**48.8±0.4**

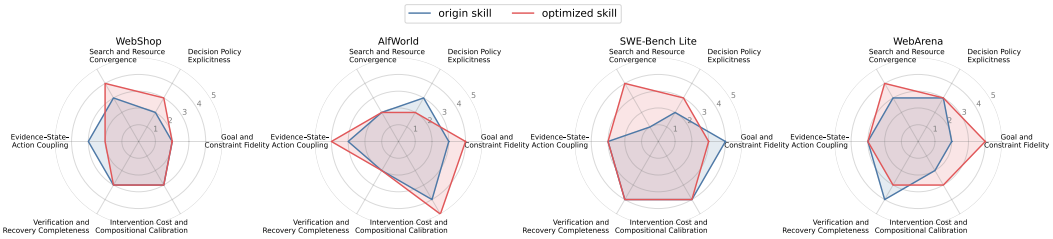


Figure 3: Skill-quality profiles before and after quality-guided optimization across the four benchmarks. Outward movement indicates higher rubric scores.

broadly consistent at the Skill level, with 43/49 and 37/49 Skills improving, respectively. These results provide complementary evidence that the optimization procedure transfers to an independently released Skill collection.

4.6 RQ5: How Robust Is SkillAlign to Design Choices?

Mapping specification. Figure 4 tests whether predictions depend on the exact capability-to-quality mapping. Replacing the secondary weights with uniform values preserves the original ranking correlations on WebShop and ALFWorld and the original judgment result on SWE-bench, indicating that these results do not rely on finely calibrated secondary weights. Under random perturbations, WebShop retains perfect rank agreement across all tested noise levels, while the ALFWorld ranking and SWE-bench judgment remain relatively stable under moderate perturbations and degrade gradually only under stronger distortions. Thus, the evaluated predictions are robust to substantial variation in secondary weights, although sufficiently large changes to the capability–quality relations can affect the results.

Table 3: Performance before and after SkillAlign-guided optimization on all 49 SWE-Skills-Bench tasks.

Method	Performance ↑	Δ	Improved	Degraded
GPT-5.4				
Original Skill	92.3 ± 0.7	–	–	–
SkillAlign-Optimized	$\mathbf{94.2 \pm 0.9}$	+1.9	43/49	6/49
GLM-5.3-Flash				
Original Skill	72.8 ± 0.9	–	–	–
SkillAlign-Optimized	$\mathbf{77.3 \pm 0.5}$	+4.5	37/49	12/49

Evaluator consistency. We next test whether the ranking behavior is specific to the evaluator used to construct the profiles. On the six WebShop variants, DeepSeek-V4-Flash, Qwen3.7-Plus (Qwen,

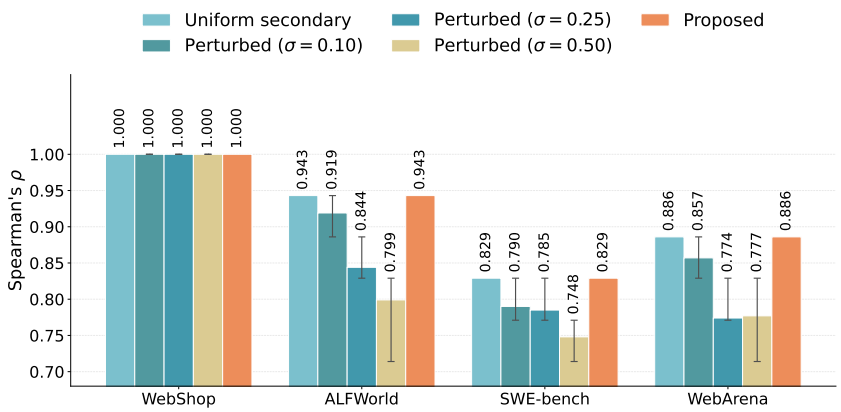


Figure 4: Robustness to capability-to-quality mapping weights. Primary assignments $p(k)$ remain fixed; perturbed results show the mean over 1,000 mappings per noise level, with error bars indicating the observed range.

2026), and Kimi-K3 (Moonshot AI, 2026) produce highly consistent rankings: pairwise Spearman correlations range from 0.943 to 1.000, Kendall correlations from 0.867 to 1.000, and overall concordance reaches $W = 0.975$. The remaining disagreement is limited to local reorderings rather than substantial changes to the overall ranking structure, suggesting that the observed ranking behavior is not specific to a single evaluator. The conclusions are also stable across evaluator models: DeepSeek-V4-Flash, Qwen3.7-Plus, and Kimi-K3 achieve high rank agreement on WebShop ($W = 0.956$) and strict HQ-selection rates of 90.0–93.3% on 30 SWE-bench pairs; full results are reported in Appendix G.2.

Hyperparameter sensitivity. Finally, we vary the primary-gate floor α, top-match weight β, task-importance exponent η, and the number of aggregated capability matches k. The representative default was fixed before this sweep and is not reselected using the reported benchmark results. Moderate variations preserve the main WebShop and ALFWorld ranking results around the default configuration, indicating that the reported behavior does not depend on a narrowly calibrated parameter setting. In particular, intermediate aggregation choices ($k = 2$ or 3) are more stable than the extremes of relying on a single capability match or aggregating all six; $\beta = 1$ is algebraically equivalent to the top-1-only ($k = 1$) ablation. Full perturbation protocols, evaluator-level results, and parameter sweeps are provided in Appendices H, G.2, and I.

4.7 Limitations

The capability and quality taxonomies and their mapping encode design judgments and may require extension in substantially different environments. Assessments depend on the evaluator and rubric, despite the consistency under evaluator and mapping variations. The benchmark profiles are descriptive, and the six-variant ranking studies and 30-pair SWE-bench analysis provide focused rather than population-level evidence. Finally, the alignment score does not capture every interaction among Skill quality, agent architecture, and environment; its agreement with performance is predictive rather than causal. Evaluation on additional benchmarks, Skills, and agent backbones is therefore needed.

5 Conclusion

We presented SkillAlign, a task-aware framework for evaluating agent skills before execution. It models task capabilities and skill qualities separately, connects them through an explicit mapping, and computes a task-specific alignment score. The same representation supports ranking, selection, quality-gap diagnosis, and optimization. The experimental design links construct validation, predictive ranking, structured ablation, optimization, downstream utility, and mapping robustness into a single evidence chain. This perspective shifts skill evaluation from asking whether a skill is good in isolation to asking whether its particular strengths match what a target task requires.

AI Assistance Disclosure

Generative AI tools were used to assist with language editing, paper organization, and drafting and revising portions of the manuscript. All AI-assisted content, analyses, and suggestions were reviewed and verified by the authors, who take responsibility for the final content of the paper.

References

Anthropic. Improving skill-creator: Test, measure, and refine agent skills. Claude by Anthropic Blog, March 2026. URL https://claude.com/blog/improving-skill-creator-test-measure-and-refine-agent-skills.

Kexin Ding, Yang Zhou, Can Jin, Feng Tong, Mu Zhou, and Dimitris N. Metaxas. Agent skill evaluation and evolution: Frameworks and benchmarks. *arXiv preprint arXiv:2606.11435*, 2026. doi: 10.48550/arXiv.2606.11435.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: Self-referential self-improvement via prompt evolution. *arXiv preprint arXiv:2309.16797*, 2023.

Haowen Gao, Haoran Chen, Can Wang, Shasha Guo, Liang Pang, Zhaoyang Liu, Huawei Shen, and Xueqi Cheng. SkillAudit: Ground-truth-free skill evolution via paired trajectory auditing. *arXiv preprint arXiv:2606.14239*, 2026. doi: 10.48550/arXiv.2606.14239.

Tingxu Han, Yi Zhang, Wei Song, Chunrong Fang, Zhenyu Chen, Youcheng Sun, and Lijie Hu. SWE-Skills-Bench: Do agent skills actually help in real-world software engineering? *arXiv preprint arXiv:2603.15401*, 2026. doi: 10.48550/arXiv.2603.15401.

Zisu Huang, Jingwen Xu, Yifan Yang, Ziyang Gong, Qihao Yang, Muzhao Tian, Xiaohua Wang, Changze Lv, Xuemei Gao, Qi Dai, Bei Liu, Kai Qiu, Xue Yang, Dongdong Chen, Xiaoqing Zheng, and Chong Luo. From raw experience to skill consumption: A systematic study of model-generated agent skills, 2026. URL https://arxiv.org/abs/2605.23899.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? *arXiv preprint arXiv:2310.06770*, 2023.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, et al. DSPy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*, 2023.

Jiaxi Li, Ke Deng, Yun Wang, Jingyuan Huang, Yucheng Shi, Qiaoyu Tan, Jin Lu, and Ninghao Liu. Online skill learning for web agents via state-grounded dynamic retrieval. *arXiv preprint arXiv:2606.04391*, 2026a. doi: 10.48550/arXiv.2606.04391.

Xiangyi Li, Yimin Liu, Wenbo Chen, et al. SkillsBench: Benchmarking how well agent skills work across diverse tasks. *arXiv preprint arXiv:2602.12670*, 2026b. doi: 10.48550/arXiv.2602.12670.

Yuan Liang, Ruobin Zhong, Haoming Xu, et al. SkillNet: Create, evaluate, and connect AI skills. *arXiv preprint arXiv:2603.04448*, 2026.

Xiao Liu, Hao Yu, Hanchen Zhang, et al. AgentBench: Evaluating LLMs as agents. *arXiv preprint arXiv:2308.03688*, 2023a.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using GPT-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023b.

Moonshot AI. Kimi K3: Think further, build bigger, keep going. Kimi, 2026. URL https://www.kimi.ai/ai-models/kimi-k3. Accessed September 25, 2026.

Jingwei Ni, Yihao Liu, Xinpeng Liu, Yutao Sun, Mengyu Zhou, Pengyu Cheng, Dexin Wang, Xiaoxi Jiang, and Guanjun Jiang. Trace2Skill: Distill trajectory-local lessons into transferable agent skills. *arXiv preprint arXiv:2603.25158*, 2026. doi: 10.48550/arXiv.2603.25158.

OpenAI. Introducing GPT-5.4. OpenAI, March 2026. URL https://openai.com/index/introducing-gpt-5-4/.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*, 2023.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*, 2023.

Qwen. Qwen3.7-Plus. Qwen Blog, 2026. URL https://qwen.ai/blog?id=qwen3.7-plus. Accessed September 25, 2026.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*, 2023.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

SkillsMP. SkillsMP: Agent skills marketplace. Online skill repository, 2026. URL https://skillsmp.com/. Accessed September 25, 2026.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, et al. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Leye Wang, Zixing Wang, and Anjie Xu. SkillTester: Benchmarking utility and security of agent skills. *arXiv preprint arXiv:2603.28815*, 2026. doi: 10.48550/arXiv.2603.28815.

Bin Xu. Ai agent systems: Architectures, applications, and evaluation, 2026. URL https://arxiv.org/abs/2601.01743.

Renjun Xu and Yang Yan. Agent skills for large language models: Architecture, acquisition, security, and the path forward, 2026. URL https://arxiv.org/abs/2602.12430.

Wei Yang, Jinwei Xiao, Hongming Zhang, Qingyang Zhang, Yanna Wang, and Bo Xu. Coarse-to-fine grounded memory for llm agent planning, 2026a. URL https://arxiv.org/abs/2508.15305.

Yutao Yang, Junsong Li, Qianjun Pan, Bihao Zhan, Yuxuan Cai, Lin Du, Jie Zhou, Kai Chen, Qin Chen, Xin Li, Bo Zhang, and Liang He. AutoSkill: Experience-driven lifelong learning via skill self-evolution. *arXiv preprint arXiv:2603.01145*, 2026b. doi: 10.48550/arXiv.2603.01145.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents. *arXiv preprint arXiv:2207.01206*, 2022a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022b.

Asaf Yehudai, Lilach Eden, Alan Li, Guy Uziel, Yilun Zhao, Roy Bar-Haim, Arman Cohan, and Michal Shmueli-Scheuer. Survey on evaluation of llm-based agents, 2026. URL https://arxiv.org/abs/2503.16416.

Dexu Yu, Youhua Li, Zhaoyang Guan, Xianhao Lin, Jining Luan, Zihao Rao, Xuanqi Lan, Yang Ran, Bo Lan, Nai-Xin Zhai, Hanwen Du, Junchen Fu, Wenhao Deng, Yongxin Ni, and Chunxiao Li. SkillAudit: From fixed-suite benchmarking to skill-centered assessment. *arXiv preprint arXiv:2606.22613*, 2026. doi: 10.48550/arXiv.2606.22613.

Z.ai. GLM-5.3-Flash: More intelligence with less compute. Z.ai Blog, September 2026. URL https://z.ai/blog/glm-5.3-flash.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, et al. Judging LLM-as-a-Judge with MT-Bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models, 2024. URL `https://arxiv.org/abs/2310.04406`.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. JudgeLM: Fine-tuned large language models are scalable judges. *arXiv preprint arXiv:2310.17631*, 2023.

A Detailed Scoring Rubrics

A.1 Evaluation Boundary and Frozen Inputs

The framework uses two independent evaluation calls. The task-profile judge receives a benchmark-level task specification and a fixed sample of task instructions, but no candidate Skill, trajectory, reward, success label, or empirical ranking. The Skill-quality judge receives one anonymized Skill and the quality rubric, but no benchmark identity, task profile, competing Skill, trajectory, reward, success label, or empirical ranking. The two frozen outputs are combined only after both judgments have completed. The same capability-to-quality matrix is used across benchmarks, tasks, models, and Skills, and all candidates within a benchmark share the same task profile and aggregation parameters.

A.2 Task-Capability Rubric

Each task capability is rated on an integer 0–5 scale. The score estimates the prospective net marginal benefit of adding a short, fixed, single-stage Skill: 0 denotes no leverage (or a stage fully handled by the environment), whereas 5 denotes maximal leverage under the frozen intervention assumption. Evaluators must cite task specifications, environment properties, available actions, or success conditions as evidence.

The task profiler receives no candidate skill, candidate score, execution trace, reward, or empirical ranking. The same frozen task profile is used for all candidates evaluated on a benchmark. When a benchmark contains materially different task families, the protocol may construct family- or instance-level profiles, but the granularity must be fixed before candidate outcomes are examined.

For each stage s, the task-profile judge assigns seven integers in $\{0, \ldots, 5\}$: decision burden A_s, branching uncertainty B_s, recurrence C_s, error amplification D_s, rule codifiability E_s, external scaffolding F_s, and guidance overhead G_s. The validator, rather than the model, recomputes

$$P_s = .25A_s + .20B_s + .10C_s + .20D_s + .25E_s, \tag{6}$$

$$N_s = \mathrm{clip}_{[0,5]}(P_s - .20F_s - .15G_s), \tag{7}$$

$$L_s = \mathrm{clip}_{\{0,\ldots,5\}}\left(\lfloor N_s + .5 \rfloor\right). \tag{8}$$

If $A_s = 0$, $N_s = L_s = 0$. The implementation rounds N_s to two decimals and orders stages by L_s, then N_s, then the fixed stage order. Each response must contain the seven factors, derived scores, an observable bottleneck event, an intervention rule, supporting and discounting evidence, a counterfactual, and confidence for every stage. Missing stages, invalid factors, missing evidence, or invalid confidence values cause rejection.

The exact system message is `Return strict JSON only.` The user prompt is generated by `run_leverage_evaluation.py::prompt_for`, which appends the complete rubric and serialized dataset input. Syntax, schema, and API failures trigger retry with capped exponential backoff.

Table 4: Task-profile evaluator configuration.

Setting	Value
Input	Task specification and fixed sampled instructions
Evaluator	DeepSeek-V4-Flash
Temperature	0
Output	Strict JSON object
Stages / factors per stage	6 / 7
Timeout / attempts	900 seconds / 3
Explicit output-token cap	None
Candidate Skills or outcomes visible	No

Table 5: Skill-quality evaluator configuration.

Setting	Value
Input	One anonymized Skill
Evaluator	DeepSeek-V4-Flash
Dimensions / range	Q_1–Q_6 / integer 0–5
Temperature	0
Output	Strict JSON object
Default workers	4
Timeout / attempts	900 seconds / 3
Task profile, competitors, or outcomes visible	No

A.3 Skill-Quality Rubric

For every quality dimension, the evaluator emits an integer score from 0 to 5. A score of 0 indicates that the relevant behavioral mechanism is absent, contradictory, or counterproductive; 3 denotes a usable intervention for ordinary cases; and 5 is reserved for an exceptional, complete, decision-local, low-risk mechanism with no substantive intrinsic weakness within scope.

Dimension-specific interpretation is as follows:

Q_1**: Goal and Constraint Fidelity.** Inspect whether goals, hard constraints, completion criteria, and priority rules are stated and preserved across steps.

Q_2**: Decision Policy Explicitness.** Inspect whether the skill connects recognizable conditions to concrete choices rather than offering only general advice.

Q_3**: Search and Resource Convergence.** Inspect prioritization, search bounds, resource selection, retry budgets, and stopping criteria.

Q_4**: Evidence–State–Action Coupling.** Inspect whether observations are recorded, updated, and used to determine subsequent action.

Q_5**: Verification and Recovery Completeness.** Inspect outcome checks, error detection, fallback strategies, and final verification.

Q_6**: Intervention Cost and Compositional Calibration.** Inspect execution burden, redundancy, conflicts with tools or other skills, and whether the guidance is appropriately scoped.

YAML front matter and version-like identifiers are removed before submission; the remaining content is unchanged. For numeric aggregation, a null dimension is replaced by that Skill's mean over applicable dimensions, or zero if none is applicable. A multi-Skill diagnostic profile is the arithmetic mean of component profiles and is not a joint evaluation of the full system prompt.

The evaluator returns the integer score, a concise rationale, supporting spans from the skill, and a list of deficiencies for each dimension. Missing evidence cannot be replaced by assumptions about what a capable model might infer. All task profiling and Skill-quality assessment use DeepSeek-V4-Flash. The exact prompts are reproduced at the end of this appendix (Appendix K).

B Fixed Mapping and Alignment Computation

The frozen mapping is

$$\mathbf{M} = \begin{bmatrix} .25 & .15 & .05 & .20 & .20 & .15 \\ .10 & .10 & .05 & .45 & .15 & .15 \\ .10 & .20 & .40 & .15 & .05 & .10 \\ .10 & .30 & .10 & .25 & .15 & .10 \\ .10 & .15 & .05 & .25 & .35 & .10 \\ .10 & .15 & .10 & .20 & .20 & .25 \end{bmatrix}. \tag{9}$$

Rows correspond to goal, state, planning, execution, verification, and adaptation; their primary dimensions are $Q_1, Q_4, Q_3, Q_2, Q_5, Q_6$.

Let $p(k)$ be capability k's primary dimension, $d_k = \sum_{j \neq p(k)} M_{k,j}$, and q_j the Skill score. The representative configuration computes

$$a_k(T) = (c_{T,k}/5)^{\eta}, \tag{10}$$

$$u_k(S) = \frac{\sum_{j \neq p(k)} M_{k,j} q_j}{d_k}, \tag{11}$$

$$c_k(S) = q_{p(k)}[\alpha + (1-\alpha)u_k(S)/5], \tag{12}$$

$$z_k(T,S) = a_k(T)c_k(S), \tag{13}$$

with $(\alpha, \beta, \eta) = (.7, .8, 1)$. If $d_k < 10^{-8}$, $u_k = 5$; this branch is not reached by the reported matrix. Sorting the six matches gives

$$F(T,S) = 20[.8z_{(1)} + .2z_{(2)}] \in [0, 100]. \tag{14}$$

The task-agnostic comparator is $F_{\text{raw}}(S) = \frac{1}{6}\sum_{j=1}^{6} q_j$.

C Quality-Guided Optimization Protocol

For each origin skill, the optimizer receives: (1) the original skill text, (2) the target-task description, (3) the quality-dimension definitions, (4) the quality scores and evidence-linked deficiencies, and (5) the ranked gap vector from Equation (5). It does not receive downstream rewards, success labels, execution traces from the evaluation set, or comparisons with the final optimized condition.

The optimizer is instructed to address the highest-priority gaps while preserving correct content and scope. Revisions should add explicit mechanisms only when they change a task-relevant decision; redundant explanations and requirements likely to interfere with other tools are discouraged.

After revision, the evaluator reassesses the optimized skill under the same rubric. No score improvement is treated as downstream evidence. Execution comparisons remain the external test of whether the proposed revision is useful.

C.1 Original and Optimized Skill Profiles

All task profiles precede candidate-Skill comparison, and all Skill profiles are produced without benchmark identity, task profiles, trajectories, or outcomes. The same frozen mapping and aggregation procedure is then applied to obtain the reported alignment scores.

D External Validation on SWE-Skills-Bench

We additionally evaluate SkillAlign on SWE-Skills-Bench, which contains 49 Skills categorized by their observed downstream effects: 7 positive, 39 neutral, and 3 negative. We apply the frozen evaluation specification without adapting the rubric, mapping, aggregation function, or hyperparameters to this benchmark.

The results provide mixed evidence of external validity. SkillAlign achieves an AUC of 0.810 when distinguishing Skills with positive downstream effects from those with negative effects, suggesting

Table 6: Quality and task–skill alignment changes averaged over all Skills in each benchmark. Quality mean is the unweighted average of the six quality scores, and A is the task-conditioned SkillAlign alignment score.

Task	Quality mean		Alignment		
	Origin	Optimized	Origin A	Optimized A	ΔA
WebShop	3.78	4.17	69.18	84.31	+15.13
ALFWorld	2.53	3.83	41.24	59.85	+18.61
SWE-bench Lite	3.41	4.33	52.94	65.82	+12.88
WebArena	3.12	3.28	46.77	62.21	+15.44

Table 7: External validation results on SWE-Skills-Bench. Positive, neutral, and negative labels are defined according to the downstream effects reported by the benchmark.

Metric	Result
Positive vs. remaining Skills AUC	0.544
Negative vs. remaining Skills AUC	0.674
Positive vs. negative Skills AUC	0.810
All-Skill Spearman's ρ	0.116

that the score captures some separation between the two outcome extremes. However, performance is substantially weaker when the neutral class is included: the positive-versus-rest AUC is 0.544, the negative-versus-rest AUC is 0.674, and the rank correlation across all 49 Skills is $\rho = 0.116$.

One likely contributing factor is the highly imbalanced and heterogeneous label distribution. Neutral Skills account for 39 of the 49 instances and may include Skills that differ in intrinsic quality, task coverage, agent compatibility, or effect size, despite sharing the same observed outcome category. The current results therefore suggest that SkillAlign distinguishes clearly beneficial from clearly harmful Skills more reliably than it distinguishes beneficial Skills from the heterogeneous neutral group. We treat this analysis as supplementary evidence rather than as support for general full-ranking validity.

E SWE-BENCH PAIRWISE CASES

The SWE-bench selection study contains 30 cases selected from the same 300-instance SWE-bench Lite set used by the downstream comparison in Table 2. An instance is eligible for the pairwise study when the agent fails without a Skill, at least one candidate Skill changes the outcome to success, and the available Skills include both successful and failed executions. These criteria focus the analysis on instances for which Skill use can affect the outcome and differences among Skills are empirically visible. The 30 cases are therefore a deliberately selected diagnostic subset, not a random or representative sample of the full 300-instance set.

For each selected instance, the pair contains a high-quality Skill associated with a successful execution (HQ) and a length-controlled low-quality Skill associated with a failed execution (LQ), evaluated under comparable backbone, harness, resource, and instance conditions. The candidate pool is not restricted to the 49 Skills in SWE-Skills-Bench: it also includes source Skills collected from public online repositories such as SkillsMP and SkillNet (SkillsMP, 2026; Liang et al., 2026). Moreover, length-control editing can produce a distinct derived LQ text for each case. The 60 entries in the 30 pairs therefore denote evaluated case-level texts, not 60 independently released Skills or a subset bounded by the 49-Skill benchmark collection. The resulting texts do not form exact length matches. Rather, the control removes a systematic length advantage for HQ: LQ is longer in 25 of the 30 pairs, by 29.7–153.8%, whereas HQ is longer in the remaining five pairs by only 3.9–5.2%. Table 8 reports the separate execution check underlying the claim that adding length alone does not change the LQ outcome. The original and expanded LQ Skills both fail all 30 tasks. SkillAlign selects the successful Skill in 28 of 30 cases (93.3%), compared with 21 of 30 (70.0%) for Raw Q Mean, 17 of 30 (56.7%) for Direct LLM, and 23 of 30 (76.7%) for the published validated rubric. Table 9 reports all case-level outcomes, length statistics, scores, and nine-call vote counts.

Table 8: Execution check before and after expanding the 30 failed LQ Skills. The expanded Skills are evaluated on the same paired SWE-bench instances under the same execution configuration.

LQ condition	Successful	Failed	Success rate
Before length expansion	0	30	0.0%
After length expansion	0	30	0.0%

Table 9: Detailed results for the 30 SWE-bench paired cases. HQ denotes the high-quality Skill associated with a successful execution, and LQ denotes the length-controlled low-quality Skill associated with a failed execution. A is the SkillAlign alignment score; $\bar{Q}$ is Raw Q Mean. The ratio is HQ characters divided by LQ characters. Vote entries are HQ/LQ counts over nine independently order-randomized calls.

Instance	Execution outcome		Length control				A		$\bar{Q}$		Votes (HQ/LQ)	
	LQ	HQ	HQ chars	LQ chars	Δ	Ratio	HQ	LQ	HQ	LQ	Direct	Rubric
django__django-11848	F	S	11,548	29,304	-17,756	0.394	88.70	51.87	4.50	3.33	5/4	3/6
django__django-12589	F	S	13,274	18,798	-5,524	0.706	74.44	50.93	4.50	3.17	9/0	9/0
django__django-13447	F	S	13,487	19,803	-6,316	0.681	86.00	40.26	4.33	2.83	9/0	9/0
django__django-13551	F	S	10,046	13,833	-3,787	0.726	88.70	34.08	4.50	2.67	5/4	9/0
django__django-14411	F	S	12,125	17,779	-5,654	0.682	86.94	41.70	4.33	3.17	9/0	9/0
matplotlib__matplotlib-22835	F	S	11,444	17,082	-5,638	0.670	86.00	32.30	4.33	2.17	9/0	9/0
matplotlib__matplotlib-25311	F	S	10,722	16,363	-5,641	0.655	71.78	69.18	4.33	4.00	9/0	9/0
mwaskom__seaborn-3190	F	S	12,093	15,686	-3,593	0.771	88.70	51.87	4.50	3.33	9/0	9/0
pylint-dev__pylint-7228	F	S	12,848	17,541	-4,693	0.732	86.00	46.97	4.33	2.67	9/0	9/0
pytest-dev__pytest-5221	F	S	10,400	14,945	-4,545	0.696	69.18	53.53	4.00	3.67	0/9	9/0
pytest-dev__pytest-6116	F	S	9,550	13,603	-4,053	0.702	83.38	30.18	4.00	2.00	9/0	9/0
pytest-dev__pytest-7220	F	S	6,921	6,580	341	1.052	89.61	34.46	4.67	2.83	9/0	9/0
scikit-learn__scikit-learn-11040	F	S	10,644	10,120	524	1.052	92.00	53.53	5.00	3.67	4/5	9/0
scikit-learn__scikit-learn-14983	F	S	11,790	16,874	-5,084	0.699	71.78	48.58	4.33	3.00	4/5	9/0
sphinx-doc__sphinx-10451	F	S	13,286	17,981	-4,695	0.739	88.70	67.73	4.50	3.67	5/4	4/5
sphinx-doc__sphinx-8506	F	S	11,383	16,870	-5,487	0.675	73.21	40.26	4.33	2.83	9/0	9/0
sympy__sympy-11400	F	S	9,550	13,754	-4,204	0.694	87.79	34.08	4.33	2.67	9/0	9/0
sympy__sympy-13915	F	S	12,661	18,539	-5,878	0.683	88.70	66.31	4.50	3.50	9/0	9/0
sympy__sympy-14817	F	S	10,645	14,436	-3,791	0.737	88.70	34.46	4.50	2.83	9/0	9/0
sympy__sympy-16281	F	S	11,783	17,636	-5,853	0.668	86.00	50.93	4.33	3.17	5/4	9/0
sympy__sympy-16792	F	S	13,081	18,323	-5,242	0.714	70.60	32.30	4.17	2.17	4/5	8/1
astropy__astropy-7746	F	S	11,200	15,400	-4,200	0.727	78.50	45.20	3.50	3.83	4/5	9/0
pallets__flask-4992	F	S	9,800	13,650	-3,850	0.718	81.38	46.97	3.67	4.00	3/6	6/3
psf__requests-3362	F	S	8,700	8,320	380	1.046	73.21	53.53	3.33	3.67	0/9	5/4
pydata__xarray-4493	F	S	12,450	17,600	-5,150	0.707	86.00	50.93	3.83	4.17	4/5	9/0
django__django-15789	F	S	10,500	13,900	-3,400	0.755	70.60	48.58	3.17	3.50	3/6	4/5
sympy__sympy-18189	F	S	9,200	12,300	-3,100	0.748	69.18	51.87	3.00	3.33	0/9	3/6
matplotlib__matplotlib-24149	F	S	11,700	15,850	-4,150	0.738	74.44	53.53	3.50	3.83	2/7	3/6
pytest-dev__pytest-8365	F	S	8,300	7,960	340	1.043	40.26	67.73	2.83	3.17	1/8	2/7
scikit-learn__scikit-learn-25500	F	S	10,100	9,720	380	1.039	34.08	69.18	2.67	3.00	3/6	4/5

Table 10: Detailed WebShop results for the six controlled Skill variants with GPT-5.4 and GLM-5.3-Flash. Higher reward and exact success are better; fewer steps are better.

Skill	Alignment	Pred. rank	GPT-5.4				GLM-5.3-Flash			
			Reward ↑	Success (%) ↑	Steps ↓	Emp. rank	Reward ↑	Success (%) ↑	Steps ↓	Emp. rank
S_1	30.54	5	61.2±0.2	32.2±0.2	10.3±2.2	5	57.8±0.2	27.9±0.2	12.6±1.3	5
S_2	49.12	3	63.1±0.7	33.2±0.1	10.5±1.4	3	58.4±0.6	28.8±0.6	11.9±0.2	3
S_3	60.9	1	64.4±0.4	33.8±0.6	9.4±1.5	1	59.2±0.2	29.6±0.5	9.8±1.9	1
S_4	46.97	4	62.8±0.6	32.4±0.4	11.3±1.9	4	58.2±0.4	28.4±0.4	11.8±0.8	4
S_5	54.1	2	63.9±0.2	33.4±0.2	9.8±2.4	2	58.7±0.3	29.3±0.7	10.3±0.7	2
S_6	24.17	6	60.9±0.7	31.8±0.4	11.2±1.7	6	56.9±0.6	27.7±0.4	12.8±1.7	6

Table 11: Detailed ALFWorld results for the six controlled Skill variants with GPT-5.4 and GLM-5.3-Flash. Higher success rate is better and fewer steps are better.

Skill	Alignment	Pred. rank	GPT-5.4			GLM-5.3-Flash		
			Success (%) ↑	Steps ↓	Emp. rank	Success (%) ↑	Steps ↓	Emp. rank
S_1	80.69	1	91.5±0.8	12.8±0.8	2	86.2±0.3	13.4±0.9	2
S_2	29.44	6	88.2±0.7	15.3±1.7	6	83.8±0.4	16.2±1.3	6
S_3	56.69	4	88.9±0.6	13.2±1.2	5	84.2±0.6	14.2±1.1	5
S_4	50	5	89.3±0.5	14.7±1.4	4	84.7±0.3	15.8±0.8	4
S_5	58.88	3	91.0±1.3	14.4±0.8	3	85.4±1.1	14.9±1.2	3
S_6	75.75	2	92.6±0.6	13.8±0.6	1	87.4±0.9	14.4±0.9	1

F Controlled-Variant Execution Details

Tables 10 and 11 provide the per-variant quantities underlying the WebShop and ALFWorld rankings in Figure 2. Alignment scores and predicted ranks are computed before execution and are therefore shared across the two agent backbones; each is shown only once per Skill. Model-specific downstream metrics are reported as mean ± sample standard deviation over four API runs, and empirical ranks are derived from the corresponding primary endpoint.

G Statistical Analysis

G.1 Ranking Agreement

Let x_i and y_i be predicted and empirical ranks for the same n skills. Spearman's correlation is the Pearson correlation of these ranks. Without ties it can be written as

$$\rho = 1 - \frac{6\sum_{i=1}^{n}(x_i - y_i)^2}{n(n^2-1)}. \tag{15}$$

Kendall's τ is additionally reported because it measures rank-order agreement. For WebShop and ALFWorld, $n = 6$ by experimental construction: the candidate set contains one controlled variant for each of Q_1–Q_6 and therefore exhausts the predefined intervention axes. In the ALFWorld experiment, the predicted ordering is

$$S_1 > S_6 > S_5 > S_3 > S_4 > S_2,$$

whereas the empirical ordering is

$$S_6 > S_1 > S_5 > S_4 > S_3 > S_2.$$

Thus, the two rankings differ by the adjacent transpositions S_1/S_6 and S_3/S_4. These swaps give $\sum_i (x_i - y_i)^2 = 4$, Spearman's $\rho = 0.8857$, and two discordant pairs out of 15, corresponding to Kendall's $\tau = 0.7333$. Equivalently, 13 of the 15 pairwise orderings are recovered. Figure 2 reports the unperturbed ranking and judgment results. The separate robustness procedure in Appendix H recomputes these predictions under perturbed mappings. Exact ties receive average ranks.

Table 12: Cross-model evaluation of the WebShop controlled Skill variants. Rank 1 denotes the highest predicted score. Absolute scores are comparable within an evaluator but are not assumed to be calibrated across evaluator models. The empirical rank is determined by downstream execution.

	Empirical	DeepSeek-V4-Flash		Qwen3.7-Plus		Kimi-K3	
Skill	Rank	Score	Rank	Score	Rank	Score	Rank
S_1	5	30.54	5	29.86	5	33.08	5
S_2	3	49.12	3	35.50	4	52.44	3
S_3	1	60.90	1	83.42	1	66.57	1
S_4	4	46.97	4	63.83	3	48.58	4
S_5	2	54.10	2	75.19	2	53.53	2
S_6	6	24.17	6	22.35	6	27.91	6

Table 13: Pairwise rank agreement between evaluator models on the WebShop controlled variants.

Evaluator pair	Spearman's ρ	Kendall's τ
DeepSeek-V4-Flash vs. Qwen3.7-Plus	0.943	0.867
DeepSeek-V4-Flash vs. Kimi-K3	1.000	1.000
Qwen3.7-Plus vs. Kimi-K3	0.943	0.867

G.2 Cross-Model Evaluator Consistency

Protocol. We examine whether the predicted Skill ranking is sensitive to the choice of evaluator model. We apply the same frozen evaluation specification to the six WebShop controlled variants S_1–S_6 using DeepSeek-V4-Flash, Qwen3.7-Plus, and Kimi-K3. Across evaluators, we keep the task and Skill inputs, scoring rubric, capability-to-quality mapping, aggregation function, and hyperparameters fixed; only the evaluator model is changed. None of the evaluators is provided with the empirical execution ranking. Because evaluator models may use different internal score calibrations, we compare the induced rankings rather than the absolute score magnitudes.

Table 12 shows that DeepSeek-V4-Flash and Kimi-K3 recover the empirical ordering exactly:

$$S_3 > S_5 > S_2 > S_4 > S_1 > S_6.$$

Qwen3.7-Plus preserves the same highest-ranked, second-ranked, and lowest-ranked variants, but exchanges the two intermediate variants S_2 and S_4:

$$S_3 > S_5 > S_4 > S_2 > S_1 > S_6.$$

Thus, all three evaluators agree on the top-1 prediction, the top-2 set $\{S_3, S_5\}$, and the lowest-ranked variant S_6.

As shown in Table 13, pairwise Spearman correlations range from 0.943 to 1.000, while pairwise Kendall correlations range from 0.867 to 1.000. Considering the three evaluator-induced rankings jointly, Kendall's coefficient of concordance is

$$W = 0.975,$$

indicating high overall rank agreement.

Relative to the empirical execution ranking, DeepSeek-V4-Flash and Kimi-K3 achieve perfect agreement ($\rho = 1.000$, $\tau = 1.000$), while Qwen3.7-Plus achieves $\rho = 0.943$ and $\tau = 0.867$. Importantly, the Qwen disagreement is confined to the relative order of two intermediate variants and does not affect the predicted best Skill.

Interpretation. The evaluator models exhibit substantial differences in absolute score calibration. For example, the predicted score of S_3 ranges from 60.90 under DeepSeek-V4-Flash to 83.42 under Qwen3.7-Plus. We therefore do not interpret cross-model differences in raw score magnitude as meaningful. Instead, the relevant output of the evaluation is the relative ordering of candidate Skills.

Table 14: Cross-model pairwise evaluation on 30 SWE-bench Skill pairs. Strict win rate is the fraction of pairs in which the evaluator selects HQ, counting ties as non-wins; non-tie win rate excludes ties, and tie-adjusted win rate assigns half credit to ties. HQ and LQ denote the Skills associated with successful and failed executions, respectively.

Evaluator	HQ	LQ	Ties	Strict	Non-tie	Tie-adjusted
DeepSeek-V4-Flash	28	2	0	93.3%	93.3%	93.3%
Qwen3.7-Plus	27	2	1	90.0%	93.1%	91.7%
Kimi-K3	28	1	1	93.3%	96.6%	95.0%

Under this criterion, the three models produce highly consistent predictions. These results provide evidence that the WebShop ranking obtained by SkillAlign is not specific to a single evaluator model, while limiting this conclusion to the evaluated WebShop variants.

We further examine evaluator dependence in the pairwise Skill-selection setting on SWE-bench. For each of the 30 cases, an evaluator compares the HQ Skill associated with a successful execution against the corresponding LQ Skill associated with a failed execution under the same frozen evaluation specification. This experiment complements the WebShop analysis: rather than measuring agreement over a complete ranking of controlled variants, it tests whether the direction of the pairwise selection result remains stable when the evaluator model is changed.

As shown in Table 14, all three evaluators strongly favor HQ. The HQ Skill is selected in 27–28 of the 30 comparisons, corresponding to strict win rates between 90.0% and 93.3%. Ties are rare, occurring in at most one case per evaluator. After excluding ties, the HQ win rate ranges from 93.1% to 96.6%; assigning half credit to ties yields win rates between 91.7% and 95.0%. Therefore, changing the evaluator model has little effect on the aggregate pairwise conclusion.

These aggregate counts establish stability of the overall selection result, but do not by themselves measure whether the evaluators disagree on exactly the same individual cases. We therefore interpret this experiment as evidence of cross-model stability in predictive performance, rather than as a claim of perfect case-level agreement.

H Mapping Robustness Procedure

We examine whether the predictive validity of SkillAlign depends on a narrowly specified capability-to-quality mapping. The analysis includes three complementary tests: local perturbation of the mapping coefficients, uniformization of the secondary weights under fixed primary assignments, and comparison against globally unstructured random mappings. These tests address different forms of sensitivity and should not be interpreted as interchangeable controls.

Local perturbation of mapping coefficients. Let $M \in \mathbb{R}^{6\times 6}$ denote the proposed capability-to-quality mapping, where each row corresponds to a capability, each column corresponds to a Skill-quality dimension, and $\sum_j M_{ij} = 1$. For perturbation level ϵ, we independently perturb the entries of each row, truncate negative values, and renormalize:

$$\widetilde{M}_{ij} = \max\{0, M_{ij} + \delta_{ij}\}, \qquad M'_{ij} = \frac{\widetilde{M}_{ij}}{\sum_h \widetilde{M}_{ih}}, \tag{16}$$

where δ_{ij} is independently sampled from the perturbation distribution associated with $\epsilon \in \{0.10, 0.25, 0.50\}$. The same perturbation distribution and scale are used for all entries at a given value of ϵ. If truncation produces a row with zero total mass, that row is resampled before normalization.

For each perturbation level, we generate 1,000 independently perturbed mappings. In every Monte Carlo draw, we recompute the predicted WebShop and ALFWorld rankings and the SWE-bench pairwise selection results reported in Figure 2. We then compare the perturbed predictions with the corresponding empirical execution outcomes. Figure 4 reports the mean and full observed range (minimum to maximum) across the 1,000 draws; the lower endpoint is therefore the worst observed ranking-agreement or selection result.

The 1,000 Monte Carlo trials concern only the mapping-robustness analysis associated with Figure 2. They are unrelated to the four independent downstream API runs summarized in Table 2. The former repeatedly perturb the evaluation mapping and recompute predictions without rerunning the downstream agents, whereas the latter are independent executions used to estimate downstream task performance.

Secondary-weight uniformization. The proposed mapping design specifies a primary quality dimension for each capability and assigns the remaining weight across five secondary quality dimensions. To test whether the predictive results depend on the precise relative allocation among these secondary dimensions, we construct a secondary-uniform mapping.

Let $p(i)$ denote the designated primary quality dimension for capability i. The secondary-uniform mapping is defined as

$$M_{ij}^{\text{sec-uniform}} = \begin{cases} M_{i,p(i)}, & j = p(i), \\ \dfrac{1 - M_{i,p(i)}}{5}, & j \neq p(i). \end{cases} \tag{17}$$

Thus, the primary assignment $p(i)$ and its original weight $M_{i,p(i)}$ remain unchanged, while the total secondary weight $1 - M_{i,p(i)}$ is distributed equally across the five secondary dimensions. Each resulting row remains normalized:

$$\sum_j M_{ij}^{\text{sec-uniform}} = 1. \tag{18}$$

This construction differs from setting every entry to $1/6$. A fully uniform matrix would remove both the primary assignment and all relative weight structure, making it unsuitable for isolating sensitivity to the allocation of secondary weights. The secondary-uniform variant instead preserves the structural primary-axis specification and changes only the relative weights assigned to the secondary dimensions.

We apply the secondary-uniform mapping to the same WebShop, ALFWorld, and SWE-bench evaluation instances used for the main predictive-validity experiment. No other component of the evaluation procedure is changed.

Random-mapping control. As a separate global control, we compare the proposed mapping against uninformative random mappings. Each row of a random mapping is sampled independently from a symmetric Dirichlet distribution:

$$M_{i,:}^{\text{rand}} \sim \text{Dirichlet}(\gamma \mathbf{1}_6), \tag{19}$$

where γ is shared across all six dimensions. The sampled rows are nonnegative and normalized by construction. Unlike the local-perturbation and secondary-uniform variants, the random-mapping control does not preserve the proposed primary capability-to-quality assignments or their numerical weight structure.

For every sampled random mapping, we recompute the WebShop and ALFWorld rankings and the SWE-bench pairwise selection result. We locate the result obtained by the proposed mapping within the resulting random-mapping distribution. Its percentile measures whether the proposed mapping performs unusually well relative to globally unstructured mappings.

The random-mapping analysis therefore addresses a different question from the perturbation analysis. Random mappings test the value of the mapping's overall structure relative to uninformative alternatives, whereas local perturbations test sensitivity to nearby numerical specifications. Secondary-weight uniformization provides an intermediate test: it preserves the primary structural assignments but removes distinctions among the secondary weights.

Results under local perturbations. Figure 4 shows that the framework is generally stable under moderate changes to the mapping coefficients, although the degree of sensitivity varies across tasks. WebShop remains invariant across the tested perturbation levels. ALFWorld and SWE-bench preserve relatively high predictive agreement under mild perturbations and degrade gradually as the

perturbation strength increases. We observe no abrupt collapse under small perturbations, indicating that the predictive results do not depend on a narrowly tuned point in the numerical mapping space.

The stronger perturbation settings should be interpreted as stress tests rather than as plausible minor variations of the proposed mapping. Because perturbation, truncation, and row normalization can collectively produce substantial redistributions of mass at large ϵ, some deterioration at the highest perturbation level is expected. The relevant observation is that degradation is gradual rather than discontinuous.

Results under secondary-weight uniformization. The secondary-uniform mapping preserves the WebShop and ALFWorld ranking results and the SWE-bench pairwise selection result obtained with the proposed mapping. Because this variant retains both the designated primary quality dimension and its original weight, the result shows that the main predictions are not sensitive to the precise relative allocation of weight among the five secondary dimensions.

This result should not be interpreted as showing that the complete mapping structure is unnecessary. In particular, the experiment does not alter the identity or weight of the primary quality dimension. It instead isolates the role of the secondary-weight allocation while holding the primary-axis structure fixed. The result is consistent with the primary-axis gate in SkillAlign, under which the designated primary quality dimension provides the main capability support and the secondary dimensions provide bounded modulation.

Interpretation. Taken together, the three analyses distinguish robustness to numerical calibration from robustness to structural specification. Local perturbations show that moderate changes to individual coefficients do not overturn the main predictive conclusions. Secondary-weight uniformization shows that the results do not rely on a particular allocation of weight among secondary dimensions when the primary assignments and weights are preserved. The random-mapping control evaluates whether the proposed structured mapping performs unusually well relative to globally unstructured alternatives.

Accordingly, the evidence supports robustness to moderate changes in mapping weights under fixed primary assignments. It does not imply that the current numerical matrix is uniquely optimal, nor does it establish that the primary capability-to-quality assignments can be freely changed. Rather, the results show that the observed predictive validity is not an artifact of narrowly tuned secondary coefficients, while sufficiently strong or unstructured changes to the mapping can reduce predictive agreement.

I Hyperparameter Sensitivity

We test whether the predictive performance of the scoring function depends on a narrowly selected hyperparameter configuration. The default setting is $(\alpha, \beta, \eta, k) = (0.7, 0.8, 1.0, 2)$. This representative configuration was fixed before the sensitivity sweep; the sweep is used only to assess robustness and is not used to select a new default from the reported benchmarks. We vary one parameter at a time over

$$\alpha \in \{0, 0.3, 0.5, 0.7, 0.9, 1\}, \qquad \beta \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1\},$$
$$\eta \in \{0.5, 1, 1.5, 2\}, \qquad k \in \{1, 2, 3, 6\},$$

while holding the remaining parameters at their default values. For $k > 1$, the highest-ranked capability receives weight β, and the remaining weight $1 - \beta$ is distributed equally across ranks 2 through k; for $k = 1$, the selected capability receives unit weight. We measure agreement between the score-induced ranking and the empirical execution ranking using Spearman's ρ and Kendall's τ.

Figure 5 shows the one-at-a-time results. On WebShop, the predicted ranking is unchanged for all tested values of η and k, and remains perfect for $\alpha \in [0.5, 0.9]$ and $\beta \in [0.6, 1.0]$. ALFWorld exhibits the same general pattern, although it is more informative because its ranking changes under extreme configurations. Its rank correlation is stable for $\alpha \in [0.5, 0.9]$ and $\beta \in [0.5, 0.9]$. The two intermediate aggregation settings, $k = 2$ and $k = 3$, outperform both selecting only the highest-ranked capability ($k = 1$) and aggregating all six capabilities ($k = 6$). Changing η over $[0.5, 2]$ does

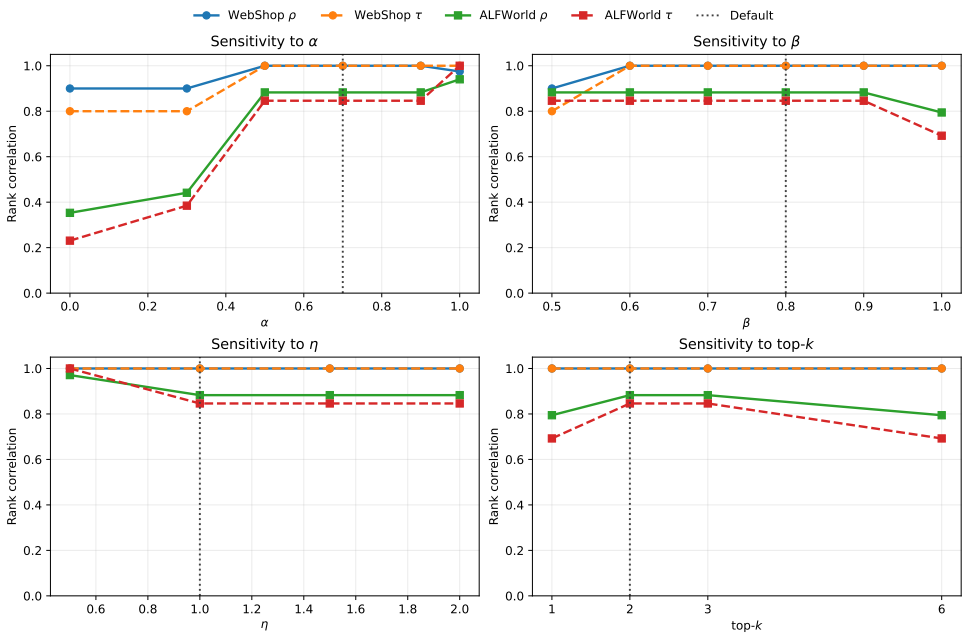


Figure 5: One-at-a-time sensitivity to α, β, η, and top-k. Each parameter is varied while the remaining parameters are held at $(\alpha, \beta, \eta, k) = (0.7, 0.8, 1.0, 2)$. Dotted vertical lines mark the default values. The default configuration lies within a broad region of stable rank correlation rather than at an isolated optimum.

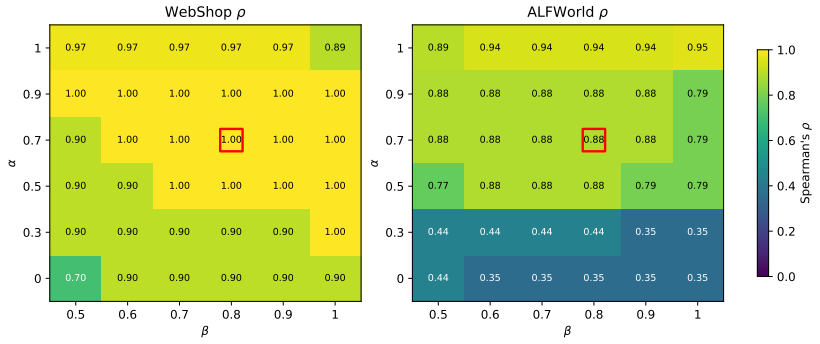


Figure 6: Joint sensitivity to α and β. Each cell reports Spearman's ρ between the predicted and empirical rankings. Red outlines mark the default setting $(\alpha, \beta) = (0.7, 0.8)$. Multiple neighboring configurations produce comparable performance on both benchmarks.

not reduce ALFWorld below the default rank correlation, with $\eta = 0.5$ producing a slightly higher correlation.

These results indicate that the main ranking conclusions do not rely on the exact numerical values of the default configuration. They also provide limited support for the top-k aggregation design: retaining two or three highly relevant capabilities is more reliable on ALFWorld than either an exclusively top-1 score or an unselective aggregation over all capabilities. Under the aggregation defined above, $\beta = 1$ assigns zero weight to every match except $z_{(1)}$ and is therefore algebraically equivalent to the $k = 1$ ablation. These two settings are consequently one structural test of top-1-only aggregation, not independent pieces of evidence.

We further evaluate all 36 combinations in the 6×6 α–β grid. As shown in Figure 6, the default configuration belongs to a contiguous high-performing region. For example, the eight configurations

$$(0.5, 0.8),\ (0.7, 0.7),\ (0.7, 0.8),\ (0.7, 0.9),$$
$$(0.9, 0.6),\ (0.9, 0.7),\ (0.9, 0.8),\ (0.9, 0.9)$$

all achieve $\rho \geq 0.99$ on WebShop and $\rho \geq 0.88$ on ALFWorld. Consequently, the reported performance cannot be attributed to selecting a single favorable combination of α and β.

Secondary-quality ablation. Setting $\alpha = 1$ removes the contribution of the secondary-quality term. This configuration obtains $\rho = 0.975$ and $\tau = 1.000$ on WebShop, and $\rho = 0.940$ and $\tau = 1.000$ on ALFWorld. It therefore does not consistently underperform the default configuration. We interpret this result conservatively: the method is robust to the inclusion and weighting of secondary-quality information, but the current ranking experiments do not establish that this term is necessary in every setting. We consequently use the sensitivity study to support robustness to parameter choice, rather than to claim that every component is indispensable.

J Reproducibility Checklist

This section specifies the frozen execution protocol and replay limitations underlying the reported experiments. The evaluator boundaries and configurations are given in Appendix A.

J.1 Additional Experimental Setup Details

WebShop is an interactive e-commerce environment in which an agent searches and navigates a simulated shopping website to purchase products satisfying natural-language requirements and attribute constraints. ALFWorld presents household tasks through a text interface grounded in embodied environments and requires multi-step navigation and object manipulation. SWE-bench Lite evaluates repository-level repair of real GitHub issues through code inspection, editing, and test-validated patches. WebArena uses realistic, self-hosted websites for long-horizon tasks involving navigation, retrieval, form interaction, and coordinated actions across applications.

The source-skill pool is assembled from publicly accessible online repositories and benchmark releases, including SkillsMP, SkillNet, SWE-Skills-Bench, and SGDR (SkillsMP, 2026; Liang et al., 2026; Han et al., 2026; Li et al., 2026a). Public provenance applies to these source Skills, not to every evaluated text. The WebShop and ALFWorld S_1–S_6 candidates are controlled variants constructed for the ranking study, and the SWE-bench LQ candidates are length-controlled derivatives constructed from source Skills. The shared DeepSeek-V4-Flash evaluator is used for task profiling, structured Skill-quality assessment and reassessment, capability-to-quality mapping assessment, and the Direct LLM and Validated Rubric baselines. Raw Q Mean and SkillAlign are computed from the same frozen task profiles and quality assessments. Holding the evaluator fixed across methods reduces evaluator-capability differences as a source of variation.

The WebShop and ALFWorld ranking studies use the complete six-variant controlled design: each S_i primarily changes one predefined quality dimension while minimizing changes to the others. This set exhausts the six intervention axes rather than subsampling a larger candidate pool. The SWE-bench downstream study uses all 300 SWE-bench Lite instances, whereas the pairwise study uses the 30 diagnostic cases selected by the criteria in Appendix E. The four-run downstream protocol is distinct from the ranking and pairwise-judgment protocols. Likewise, the 1,000 mapping perturbations at each noise level in Appendix H concern robustness of the pre-execution predictions and are not additional downstream API runs.

Benchmark-specific outcomes are WebShop reward, exact success rate, and average steps; ALFWorld success rate and average steps; SWE-bench Lite resolved rate; and WebArena task success rate. Assessment protocols are fixed before the corresponding outcomes are analyzed.

J.2 Baselines and Statistical Evaluation

We compare SkillAlign with three pre-execution baselines. Direct LLM performs a holistic pairwise judgment without the structured quality decomposition. Validated Rubric applies the published

Table 15: Execution settings for downstream evaluation and pairwise selection. A dash indicates that a step cap is not applicable.

Experiment	Evaluation set	Repetitions	Step cap	Primary endpoint
WebShop downstream comparison	fixed 500 tasks	4 API runs	30	mean reward
ALFWorld downstream comparison	134 tasks	4 API runs	30	success rate
SWE-bench pairwise selection	30 matched pairs	9 votes/judge/pair	–	accuracy
SWE-bench downstream comparison	300 instances	4 API runs	–	resolved rate
WebArena downstream comparison	764 single-domain tasks	4 API runs	30	success rate

structured rubric of Huang et al. (2026). Raw Q Mean is the unweighted, task-agnostic average in Equation (4). Raw Q Mean and SkillAlign use the same frozen Skill-quality assessments and differ only in how those scores are matched to task requirements and aggregated.

For model-based pairwise judgments, the evaluators receive anonymous candidate Skills and the task description but not success labels, execution traces, rewards, empirical rankings, or the competing methods' scores. Direct LLM and Validated Rubric each make nine calls per SWE-bench pair with independently randomized A/B presentation order; majority vote determines the selected Skill.

Ranking agreement is reported using Spearman's ρ and Kendall's τ. Pairwise selection is reported as accuracy with 95% Clopper–Pearson exact binomial intervals. Matched comparisons between SkillAlign and each pairwise baseline use two-sided exact McNemar tests with Holm correction. Downstream results are reported as mean $\pm$ sample standard deviation over four API runs, and matched-instance differences are assessed with paired bootstrap confidence intervals.

J.3 Benchmark Execution Protocol

The four downstream runs are separate executions with identical settings and temperature zero, without explicitly controlled random seeds. Task identifiers are retained for paired comparisons. WebShop uses paired reward differences; ALFWorld uses binary success.

An earlier optimized-Skill follow-up reran only the 90 failures under the original Skill condition. That artifact measures conditional recovery and cannot estimate a full 300-instance net effect because original successes were not rerun.

J.4 Artifact Integrity and Replay Limitations

Analysis directories retain manifests, raw or cached responses, validated JSON, and reports; selected frozen inputs and outputs are indexed by SHA-256. Cached offline recomputation can be exact, whereas fresh provider calls need not be byte-identical because provider revisions and access dates are not uniformly frozen. The release also lacks one immutable dependency lock or container image covering every benchmark. Missing provider-internal revisions must therefore be reported as unknown rather than inferred from product names or file timestamps.

K Exact Evaluation Prompts

K.1 Exact Task-Profile Prompt

```
# Task–Skill Stage Leverage Rubric

## Six Stage Boundaries

1. `goal_modeling`: Extracting goals, hard constraints, quantities, ordering, and success
↪  conditions.
2. `state_modeling`: Maintaining current facts, history, inventory, counts, explored areas,
↪  and unknowns.
```

```
3. `planning`: Choosing the next step, arranging search/phases, setting budgets, and setting
↪  stop/switch conditions.
4. `execution`: Turning a decided action into a legal call, including syntax, parameters, and
↪  preconditions.
5. `verification`: Obtaining and interpreting evidence, checking items one by one, and
↪  deciding whether to conclude.
6. `adaptation`: Diagnosing, backing off, and choosing an alternative route after failure or
↪  a falsified assumption.
```

```
## Seven Sub-Scores (integers 0–5)
```

```
### Positive Items

#### A. `decision_burden` -- Unconstrained Decision Burden

0 means the stage does not exist or is fully automatic; 5 means the stage repeatedly
↪  requires autonomous choice, with task rules not directly specifying the correct action.

#### B. `branching_uncertainty` -- Branching and Uncertainty

0 means there is a single obvious path; 5 means there are many candidates, partial
↪  observability, an easy tendency to loop, and a clear need for explicit
↪  choose/stop/switch rules.

#### C. `recurrence` -- Exposure Frequency

0 means it never occurs; 1 means at most once per task; 3 means it occurs several times in
↪  most tasks; 5 means it recurs almost every step or across multiple phases.

#### D. `error_amplification` -- Error Amplification

0 means no impact; 5 means early errors propagate silently, exhaust the budget, or cause
↪  irreversible failure. Scores are lowered when errors can be spotted immediately and
↪  retried cheaply.

#### E. `rule_codifiability` -- Fixed-Rule Codifiability

0 means it is highly dependent on case-by-case knowledge and cannot be improved by general
↪  written rules; 5 means a small number of explicit gates, checklists, boundaries, or
↪  recovery rules can cover most tasks.

### Negative Items

#### F. `external_scaffolding` -- Environmental Scaffolding

0 means the environment provides no help; 5 means the environment already handles this stage
↪  for the agent through optional actions, complete state, reliable error/success signals,
↪  or a fixed process. Points should only be deducted for scaffolding that already exists
↪  for the current stage.

#### G. `guidance_overhead` -- Standing Guidance Burden

0 means the rule is short, event-triggered, and takes up almost no attention; 5 means it
↪  requires maintaining a long record every step, repeating environment information, or
↪  performing extra actions, which is likely to crowd out normal reasoning.
```

K.2 Exact Skill-Quality Prompt

```
# Behavioral Intervention Utility Rubric v9
```

```
## Purpose and evaluation boundary
```

```
Evaluate the Skill as a proposed intervention on an Agent's behavior, not as a document. The
↪  central question is whether the text can reliably change a consequential decision or
↪  action with acceptable cost.
```

```
Judge only the anonymous Skill text. Candidate identity, dataset, task profile,
↪  trajectories, rewards, empirical ranking, and other Skills are unavailable. Do not infer
↪  that a particular environment or baseline has a known failure. The downstream task
↪  profile supplies task-level leverage; this rubric supplies an intrinsic
↪  intervention-capability vector that uses the same Q1-Q6 meanings as the frozen
↪  stage-to-Q matrix.

Do not reward headings, length, polished prose, examples, exhaustive scope disclaimers,
↪  field count, or local completeness by themselves. Credit them only when they make a
↪  behavioral intervention more reliable. Do not penalize a narrow Skill for omitting
↪  out-of-scope stages.

Use integer scores only: 0, 1, 2, 3, 4, or 5. Never emit a fractional score. Use
↪  `not_applicable` with a null score only when the dimension genuinely cannot arise within
↪  the declared responsibility; weak relevance is not non-applicability.
```

Required scoring process

```
For every dimension:

1. Identify the intervention chain supported by the text:

   `available evidence -> predicate/state -> changed judgment or action -> intended avoided
   ↪  error or completed outcome`

2. Separate three concepts:
   - **specification presence**: the mechanism is written down;
   - **behavioral directness**: the mechanism actually constrains or selects a
   ↪  decision/action;
   - **adherence feasibility**: an Agent can apply it at the relevant decision point without
   ↪  fragile reconstruction or excessive competing rules.
3. Score using that dimension's own 0-5 anchors. Do not substitute a generic quality scale.
4. For every score below 5, report the dominant gap and classify it with:
   - `gap_frequency`: `rare`, `occasional`, or `common` within the declared responsibility;
   - `gap_severity`: `minor`, `material`, or `blocking` for the intervention chain;
   - `gap_scope`: `edge_case`, `recovery`, or `main_path`;
   - `primary_issue_id`: a short stable identifier.
5. For score 5 or `not_applicable`, use `none` in every gap-classification field.
```

Calibration rules

```
- Score 3 means a usable intervention for ordinary cases, not poor quality.
- Score 4 requires a direct and realistically followable mechanism across normal and
↪  important exception cases. Merely having one small-looking textual gap is insufficient.
- Score 5 is exceptional: the intervention chain is complete, decision-local, low-risk, and
↪  has no substantive intrinsic weakness within scope.
- A detailed Skill may score 1 or 2 if it mainly records state, repeats obvious environment
↪  constraints, or adds rules without changing a decision.
- A short Skill may score 4 or 5 if it installs a precise, high-leverage, low-cost decision
↪  rule.
- Do not force a score distribution and do not compare the candidate with imagined
↪  alternatives.
```

Cross-dimension ownership and duplicate penalties

```
Assign a weakness first to the dimension that directly owns it. Reuse the same
↪  `primary_issue_id` in another dimension only when the same text causes a distinct
↪  consequence there, and state that consequence explicitly.

- lost or weakened goal conditions -> primarily Q1;
- missing choice, priority, or alternative action -> primarily Q2;
- unbounded search, retries, or resource use -> primarily Q3;
- evidence or state without a decision consequence -> primarily Q4;
- unhandled failure, terminal error, or incomplete recovery -> primarily Q5;
- recurring burden, false blocking, stale state, or coordination risk -> primarily Q6.
```

```
Do not lower all dimensions because one phrase is vague. For example, an undecidable
↪  completion predicate may affect Q4 if evidence cannot determine the judgment and Q5 if
↪  it blocks closure; it does not automatically reduce Q1-Q3 and Q6.
```

```
## Exact and uncertain predicates
```

```
Judge the underlying test, not vocabulary such as `satisfy`, `match`, or `complete`.

- **Exact observable**: computable from declared visible/current evidence, such as visible
↪  price `< limit`, selected value equals requested value, two distinct identifiers are in
↪  inventory, or a named confirmation is visible.
- **Evidence-incomplete**: individual checks are possible, but missing, ambiguous, or
↪  conflicting evidence has no rule.
- **Subjective/unavailable**: `plausible`, `reasonable`, `best`, or `appears suitable` lacks
↪  an operational comparator, or the judgment requires hidden facts.

Exact observable gates can be strong positive interventions. Penalize uncertain gates
↪  according to their behavioral consequence, not by applying a mechanism-wide automatic
↪  cap.
```

```
## Q1 Goal preservation and requirement control
```

```
Evaluate whether the Skill prevents consequential goal information from being lost,
↪  weakened, silently changed, or used inconsistently across decisions. Relevant
↪  information includes identity, hard constraints, quantities, strict comparisons,
↪  required state transformations, destinations, variants, and prerequisite order. A record
↪  earns credit only if it is stable or consumed at a relevant decision/handoff.

- **0**: The Skill does not preserve goal conditions, or a central rule explicitly discards
↪  or contradicts them.
- **1**: It reminds the Agent to consider the goal but provides no reliable representation,
↪  preservation rule, or decision use; common goal loss remains unchecked.
- **2**: It preserves one or two obvious conditions, but important constraint types,
↪  hard/soft distinctions, quantities, order, or update rules are missing; material loss
↪  can occur on ordinary multi-constraint cases.
- **3**: It preserves the main goal and hard conditions for normal cases, but one important
↪  ambiguity, conflict, persistence, handoff, or re-check rule remains indirect.
- **4**: Goal-critical fields and non-relaxable conditions are explicitly represented and
↪  consumed at major decision points; missing/ambiguous evidence and allowed relaxation are
↪  mostly controlled, with only rare or minor residual risk.
- **5**: All declared goal-bearing information is represented at the needed precision,
↪  preserved across retries and handoffs, checked before consequential decisions, and
↪  protected against silent relaxation, contradiction, quantity loss, unit loss, and order
↪  violation without unnecessary maintenance.

Do not give high Q1 merely because the responsibility boundary is clear. Conversely, a Skill
↪  focused on another mechanism can legitimately receive a low Q1 without being generally
↪  poor.
```

```
## Q2 Decision policy specificity and behavioral delta
```

```
Evaluate whether the Skill installs a concrete selection policy: priorities,
↪  allowed/forbidden actions, conflict resolution, switching conditions, or alternative
↪  action classes. The policy must do more than restate a goal or enumerate legal actions.
↪  Since the actual baseline is unavailable, judge intrinsic behavioral delta: would
↪  following the text choose, forbid, or prioritize an action that a generic goal statement
↪  leaves open?

- **0**: No decision policy exists, or central rules prescribe mutually incompatible actions
↪  in the same state.
- **1**: The text states desired outcomes or general advice but leaves the actual choice
↪  unconstrained; it adds no stable behavioral delta.
- **2**: At least one choice is constrained, but priorities, conflicts, forbidden-case
↪  alternatives, or key predicates are missing; ordinary cases can still branch
↪  arbitrarily.
- **3**: The normal decision policy selects or forbids actions in observable situations, but
↪  one material priority, tie, uncertainty case, or alternative route remains implicit.
```

```
- **4**: Normal and major exception decisions have explicit priorities and alternatives;
↪  prohibitions route to a different useful action class; remaining ambiguity is rare and
↪  unlikely to change the main choice.
- **5**: Every consequential in-scope choice has an observable policy, ordered priorities,
↪  conflict handling, and a productive alternative when blocked; the rules create a clear
↪  behavioral delta without over-specifying irrelevant actions.

Do not award high Q2 for a comprehensive action-syntax list unless it decides among
↪  competing actions or changes what is permitted in a consequential state.
```

Q3 Search, retry, and resource convergence

```
Evaluate whether the Skill makes potentially repetitive work converge while preserving
↪  useful exploration. Consider queries, candidates, locations, pagination, retries,
↪  evidence checks, state reviews, and finite action budgets. Convergence requires both a
↪  progress test and a productive response to no progress; a counter or hard limit alone is
↪  insufficient.

- **0**: The Skill creates or mandates an unbounded loop, repeated action, or resource
↪  policy that cannot terminate.
- **1**: It mentions efficiency or avoiding repetition but supplies no observable progress
↪  state, novelty test, bound, switch, or exhaustion behavior.
- **2**: It controls one repetition mode, but common loops remain possible, progress is
↪  subjective, or stopping merely abandons the task without a useful next route.
- **3**: Major normal loops have observable novelty/progress checks and termination or
↪  switching rules, but a material retry class, budget tradeoff, false-stop risk, or escape
↪  condition remains weak.
- **4**: Search and retry behavior is bounded by observable history or progress; no-progress
↪  causes a materially different action; useful exploration is preserved through justified
↪  escape rules; remaining resource risk is rare or minor.
- **5**: All in-scope repetitive paths have minimal sufficient state, exact novelty/progress
↪  tests, calibrated bounds or feedback-gated switches, productive exhaustion behavior, and
↪  explicit protection against both futile repetition and premature cutoff.

Do not credit a general state ledger merely because it contains visited items. Credit only
↪  the fields that actually prevent repetition, allocate resources, or trigger a different
↪  action. Do not automatically penalize hard limits; judge false-stop likelihood,
↪  reversibility, and escape behavior.
```

Q4 Evidence-to-state-to-action coupling

```
Evaluate whether declared inputs, observations, or feedback cause explicit changes in a
↪  decision-relevant state, judgment, output, handoff, or next action. Use the audit chain
↪  `field -> predicate -> consumer -> consequence`. A narrow producer may hand off rather
↪  than act, but the consumer and decision consequence must be declared.

In the evidence, report a `verb_composition` audit with counts of enumeration, adjudication,
↪  and conditional-mapping rule units. Counts are diagnostic only and must not mechanically
↪  determine the score.

- **0**: Evidence is mentioned or collected but has no stated in-scope consequence.
- **1**: The Skill mainly enumerates, copies, or stores observations; a consumer or action
↪  effect is merely implied.
- **2**: Some evidence changes a judgment or action, but most critical fields lack
↪  predicates or consumers, or the main intervention depends on an unconsumed record.
- **3**: Main-path evidence drives a defined judgment, handoff, or action, but a material
↪  missing/ambiguous evidence case, conflict, update rule, or alternative consequence
↪  remains unspecified.
- **4**: Critical normal and major exception evidence has explicit predicates and consumers
↪  that change state, judgment, handoff, or action; retained fields are decision-relevant
↪  and residual gaps are rare or minor.
- **5**: Every critical evidence state, including missing, conflicting, unchanged, and
↪  failure feedback, maps through explicit update/priority rules to a consequential
↪  decision or typed handoff; no maintained field lacks a justified consumer.

Do not reward more fields. A single exact signal that reliably changes the next action may
↪  be stronger than a complete persistent ledger with weak consumption.
```

Q5 Consequential error prevention, recovery, and closure

```
Evaluate whether the Skill prevents high-impact local errors before they occur, recovers
↪  from diagnostic failure, and reaches an explicit success, safe failure, termination, or
↪  handoff. Local completeness matters only insofar as it protects the outcome; editorially
↪  complete lifecycle prose is not itself valuable.

- **0**: The main path cannot reach the declared outcome, or the Skill directs a known
↪  failure without recovery or handoff.
- **1**: A favorable path exists only in fragments; a common failure, irreversible decision,
↪  or completion condition is unguarded and commonly blocks the responsibility.
- **2**: The normal path can complete, but a material local error has no pre-action guard or
↪  productive recovery; failure handling repeats, stops arbitrarily, or delegates without a
↪  usable signal.
- **3**: Normal closure and some major failures are handled, but one important
↪  irreversible-action check, diagnostic branch, retry exhaustion, ambiguous-evidence case,
↪  or handoff remains incomplete.
- **4**: Consequential normal-path errors are checked before commitment; major diagnostic
↪  failures map to corrected retries, alternative routes, explicit failure, or typed
↪  handoff; only rare or low-impact closure gaps remain.
- **5**: Every important in-scope error opportunity has a decision-local guard or
↪  evidence-based recovery; terminal actions use exact observable conditions; retries are
↪  bounded; exhaustion produces a safe explicit outcome; the mechanism neither falsely
↪  claims success nor strands a correct path.

A block-only gate is weaker than a gate with a productive alternative, but a narrow
↪  verification Skill may legitimately hand off a typed failure rather than choose the
↪  route itself. Exact option, quantity, price, inventory, or confirmation checks are
↪  positive evidence, not fuzzy gates.
```

Q6 Net intervention utility, adherence feasibility, and interference

```
Evaluate the intrinsic benefit-cost profile of following the Skill: decision locality, rule
↪  salience, history and state burden, false triggers, false blocks, stale-state
↪  propagation, repeated checks, rule competition, and coordination dependencies. Do not
↪  guess actual reward or baseline failure frequency; assess whether the intervention is
↪  plausibly realizable and whether its costs are proportionate to the behavior it
↪  controls.

- **0**: The intervention predictably defeats its own responsibility through deadlock,
↪  pervasive false blocking, contradictory priorities, or overwhelming recurring work.
- **1**: Main-path adherence is fragile or commonly harmful because critical rules require
↪  unavailable history, excessive maintenance, repeated unnecessary actions, or have no
↪  escape from false triggers.
- **2**: The mechanism can help, but recurring burden, stale-state risk, duplicated
↪  observation, hard-control interference, or coordination dependence is material and often
↪  competes with the intended behavior.
- **3**: The intervention is usable and costs are broadly proportionate, but one important
↪  maintenance policy, history dependency, false-block risk, priority conflict, or
↪  implementation-placement assumption remains weak.
- **4**: Key rules are salient at the relevant decision point, use minimal sufficient state,
↪  and are realistically followable; recurring work has a consumer; false triggers and
↪  blocking have safe bounds; remaining interference is rare or minor.
- **5**: The intervention is compact relative to its responsibility, event- or
↪  change-triggered where possible, decision-local, robust to stale or missing state,
↪  explicit about priorities and escape paths, and introduces no substantive recurring,
↪  interference, or coordination cost.

Apply these distinctions:

- One-time goal extraction may be low cost when later checks consume it.
- Failure-triggered recovery may be low cost when dormant on successful paths and feedback
↪  is diagnostic.
- Per-observation maintenance is costly when fields duplicate visible state, require
↪  long-history reconstruction, or can propagate an incorrect record.
- A globally coherent controller can be easier to follow than the same contract scattered
↪  across many local responsibilities; within a single-file judgment, credit explicit
↪  locality and declared shared-state/handoff assumptions, but do not invent unseen
↪  modules.
- More verification is not automatically safer when it causes repeated navigation or
↪  evidence collection without changing a decision.
```

Final safeguards

```
- Score intrinsic intervention capability, not observed performance.
- The task profile, not this judge, determines which stage is valuable for a specific task.
- A low score in an out-of-focus dimension does not make a narrow Skill bad; it produces the
↪  capability shape needed for downstream matching.
- Do not make all competent Skills score 4 in every dimension. A specialized intervention
↪  should normally show peaks in the mechanisms it actually implements and lower scores
↪  elsewhere.
- Do not infer that a named object, action, or environment has a particular empirical
↪  frequency.
- Do not reward a Skill for claiming that it targets a baseline failure; require an
↪  operational mechanism in the body.
```