ROSE: ROLLOUT-FREE SKILL EVOLUTION VIA STATIC TASK–SKILL ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Skill evolution offers a practical way to improve agent performance without updating model weights. Existing methods often rely on repeated task rollouts and evaluator feedback to guide skill revision, making skill construction costly and time-consuming. Yet procedural deficiencies, such as formatting errors and omitted steps, can often be identified from task requirements before execution. Motivated by this observation, we introduce ROSE, a rollout-free framework for efficient agent skill evolution inspired by static program analysis. ROSE organizes an initial skill, training-task specifications, and visible artifact metadata into a shared code-style intermediate representation. It then audits this representation for unmet requirements, contract mismatches, and predictable execution risks, and uses the findings to guide localized skill revisions while preserving useful procedures. Construction requires only three LLM calls, without task rollouts or evaluator feedback. The resulting skill is frozen and reused across held-out tasks without test-time adaptation. Across SpreadsheetBench, LiveMathematicianBench, DocVQA, and OfficeQA with four agent models, ROSE achieves competitive performance, obtaining a macro-average score of 58.5%, compared with 56.8% for SkillOpt. On matched SpreadsheetBench construction instances, ROSE uses 9,183 tokens and completes construction in 32.7 seconds, compared with 3.86 million tokens and approximately 100 minutes for SkillOpt. These results demonstrate that static task evidence can support competitive held-out performance with substantially reduced skill-construction cost. The code will be publicly available.

1 INTRODUCTION

Language-model agents increasingly rely on external skills that encode reusable procedures and task-solving guidance. These skills support a wide range of applications, including tool use (Schick et al., 2023; Yao et al., 2022), embodied interaction (Shridhar et al., 2020; Ahn et al., 2022), web navigation (Zhou et al., 2024), software engineering (Yang et al., 2024), and long-horizon reasoning (Shinn et al., 2023). As agents tackle increasingly complex tasks involving mathematical reasoning, coding, and interaction with dynamic environments, skill quality becomes a critical determinant of performance: even a capable underlying model may fail when guided by an incomplete, ambiguous, or otherwise poorly specified skill.

Existing approaches address skill deficiencies primarily through feedback-driven optimization. Earlier work applied general-purpose text optimizers, such as TextGrad (Yuksekgonul et al., 2024) and GEPA (Agrawal et al., 2025), to skill revision using natural-language feedback and candidate selection. More recent work has developed skill-specific methods that distill execution trajectories, translate failures into textual gradients, or perform bounded edits gated by validation outcomes (Ni et al., 2026; Wang et al., 2026; Yang et al., 2026). Across this progression, task execution remains central to revealing skill deficiencies and generating signals for revision or selection. Consequently, skill construction requires repeated rollout-and-evaluation cycles, making it computationally expensive and time-consuming in practice.

However, we observe that many agent failures stem from basic formatting errors or procedural omissions rather than difficulties with complex reasoning. For example, an agent may return an answer in the wrong format, mishandle an output file, or omit a required completion step. Such failures often reflect mismatches between task requirements and the procedures specified in a skill. These

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.

mismatches can often be identified and addressed before execution by examining training-task specifications, visible artifacts, and the initial skill. This observation suggests that useful skill updates can be derived directly from static task evidence, without incurring the substantial costs of task rollouts and evaluator feedback collection during skill construction.

Motivated by this observation, we introduce ROSE (**RO**llout-Free **S**kill **E**volution), a rollout-free framework for agent skill evolution inspired by static program analysis. ROSE maps the initial skill and training-task evidence into a shared code-style intermediate representation, enabling direct comparison between task requirements and skill procedures. It then audits this representation for unmet requirements and predictable execution risks, and uses the findings to synthesize a localized skill update. The entire construction process requires neither task rollouts nor evaluator feedback.

Specifically, the intermediate representation (IR) organizes skill procedures, task requirements, and artifact metadata into aligned typed records. Task specifications describe required operations, output formats, and completion criteria, while artifact metadata captures the structure and properties of relevant files, tables, and other task objects. ROSE audits these records for missing interfaces, contract mismatches, and predictable execution risks. The audit findings guide a localized synthesis stage that preserves valid procedures while adding missing guidance, clarifying ambiguous instructions, and introducing bounded recovery rules where needed. The resulting skill is frozen before evaluation and reused unchanged across held-out tasks.

Experiments across four benchmarks and four agent models demonstrate that ROSE achieves competitive held-out performance using only three LLM calls for skill construction. It obtains a macro-average score of 58.5%, compared with 56.8% for SkillOpt, with particularly strong results on SpreadsheetBench: 61.0%, compared with 50.0% for SkillOpt and 34.8% without skills. On matched SpreadsheetBench construction instances, ROSE requires only $9{,}183$ tokens and 32.7 seconds, compared with 3.86 million tokens and approximately 100 minutes for SkillOpt. These findings demonstrate the potential of static task evidence to support competitive held-out performance with substantially reduced skill-construction cost.

The contributions are summarized as follows:

- We introduce a static program-analysis perspective on skill evolution, using task specifications and visible artifact metadata to identify procedural deficiencies before execution.
- We propose ROSE, a rollout-free framework that combines code-style IR construction, programmatic skill auditing, and localized skill synthesis to produce reusable skill updates with only three LLM calls, without task rollouts or evaluator feedback.
- Experiments across four benchmarks and four agent models demonstrate competitive held-out performance, while showing substantial reductions in token usage and skill-construction time.

2 Related Work

In this section, we position ROSE relative to two closely related lines of research: LLM and agent skills and agent skill evolution.

LLM and agent skills. Recent work increasingly treats skills as persistent procedural artifacts (Xu & Yan, 2026; Jiang et al., 2026; Zhou et al., 2026). Existing studies examine public skill ecosystems (Ling et al., 2026), skill acquisition and reuse (Chen et al., 2026), structured or executable skill representations (Liu et al., 2026; Li et al., 2026b), and large-scale skill organization and compilation (Li et al., 2026a; Pan et al., 2026). These studies primarily investigate how skills are represented, acquired, organized, and reused. In comparison, ROSE focuses on improving existing skills by identifying procedural deficiencies through static comparisons with training-task requirements and visible artifact structures.

Agent skill evolution. Existing skill-evolution methods primarily improve textual skills using execution experience. Trace2Skill distills lessons from execution trajectories into transferable skill patches (Ni et al., 2026), SkillGrad converts failed executions into textual gradients (Wang et al., 2026), and SkillOpt performs bounded optimization through rollout-based evaluation and validation-guided editing (Yang et al., 2026). Broader textual optimization and self-evolving frameworks use

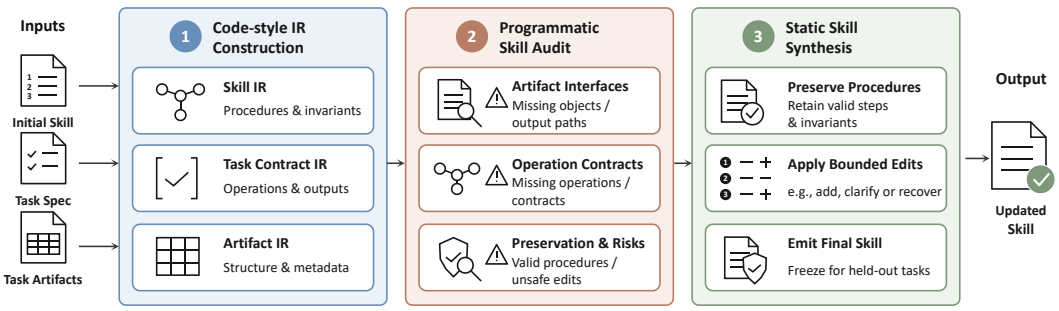

Figure 1: Overview of the ROSE pipeline. Training-task specifications, and visible artifact metadata provide the static evidence for skill construction. Given this evidence and an initial skill, ROSE constructs a code-style IR, jointly audits procedural deficiencies and predictable risks, and synthesizes an updated skill. Each stage uses one LLM call. Internal boxes denote IR components, audit perspectives, or synthesis requirements. The updated skill is frozen and reused across held-out tasks without construction rollouts or evaluator feedback.

natural-language feedback, trajectory analysis, reinforcement learning, or repeated candidate evaluation to guide updates (Yuksekgonul et al., 2024; Agrawal et al., 2025; Vishe et al., 2026; Ma et al., 2026; Tian et al., 2026). In contrast, ROSE derives skill updates through one-pass static analysis of the initial skill, training-task specifications, and visible artifacts, requiring neither task rollouts nor evaluator feedback during skill construction.

3 Methodology

ROSE revises a reusable skill through three sequential stages: *Code-style IR Construction*, *Programmatic Skill Audit*, and *Static Skill Synthesis*. The first stage organizes the initial skill and permitted static evidence into a shared structured representation. The second conducts a joint audit to identify procedural deficiencies, predictable execution risks, and useful existing procedures to preserve. The third uses the resulting audit report to guide concise, evidence-supported revisions to the skill document. As illustrated in Figure 1, each stage requires a single LLM call. The updated skill is then frozen and reused unchanged across held-out evaluation tasks. Detailed construction prompts and a complete example of the three-stage workflow are provided in Appendix 8.

3.1 Problem Formulation

We represent a human-readable skill directory as:

$$S = (\mathcal{M}, \mathcal{R}), \tag{1}$$

where $\mathcal{M}$ is the root `SKILL.md` document and $\mathcal{R}$ contains auxiliary references, scripts, or assets. Given a fixed LLM agent π_θ and an initial skill S_0, the goal of skill evolution is to construct an updated skill S^* such that:

$$\mathcal{P}(S^*; \pi_\theta, D_{\text{test}}) > \mathcal{P}(S_0; \pi_\theta, D_{\text{test}}), \tag{2}$$

where D_{test} denotes the held-out test set and $\mathcal{P}(S; \pi_\theta, D_{\text{test}})$ measures the agent's task performance when using skill S under the corresponding benchmark's evaluation metric. The resulting updated skill is frozen before the start of held-out evaluation.

3.2 Code-style IR Construction

The initial skill, task specifications, and visible artifact descriptions differ in structure and level of abstraction, making direct comparison difficult. Inspired by static program analysis, ROSE maps these inputs into a shared code-style intermediate representation (IR) comprising three complementary components: Skill IR, Task Contract IR, and Artifact IR, as illustrated in the first stage of Figure 1. The IR organizes procedural guidance and task evidence into structured, non-executable records, making task requirements directly comparable with the procedures specified in the initial skill. This

representation supports the subsequent audit of artifact interfaces, operation contracts, preservation requirements, and predictable execution risks.

Evidence boundary. Skill construction uses the initial skill, training-task specifications, visible artifact metadata, and tool-execution constraints specified in the construction prompts. Together, these inputs describe the available tool interfaces, required output formats, and relevant environment semantics. All inputs are analyzed statically, without executing training tasks or evaluating candidate skills during construction. Ground-truth answers, rewards, reference solutions, execution trajectories, and evaluator feedback are excluded from the construction evidence. The process produces a single updated skill, which is frozen before held-out evaluation and reused unchanged across test tasks. Neither held-out tasks nor evaluation outcomes are used for skill construction or selection.

IR schema filling. The construction model ϕ jointly generates three complementary IR components under fixed schemas in a single LLM call. Let $Z = (Z_s, Z_t, Z_a)$ denote the resulting structured representation, where Z_s, Z_t, and Z_a correspond to the Skill IR, Task Contract IR, and Artifact IR, respectively. We formulate this construction stage as:

$$Z = \mathrm{IR}_\phi(S_0, T_{\mathrm{train}}, A_{\mathrm{train}}), \tag{3}$$

where S_0 is the initial skill, T_{train} contains training-task specifications, and A_{train} provides visible artifact descriptions and metadata. The ***Skill IR*** Z_s captures skill procedures, invariants, and preservation rules. The ***Task Contract IR*** Z_t specifies required operations, output contracts, and completion conditions. The ***Artifact IR*** Z_a records artifact structure, relevant locations, and potential hazards. Together, these components provide complementary views of the construction context, making relationships between available procedures, task requirements, and artifact properties explicit. Schema fields are populated using facts explicitly stated or directly supported by the permitted evidence. When the available information does not support a concrete field value, the construction prompt requests that it remain unspecified. This explicit treatment of incomplete evidence helps limit unsupported assumptions while preserving the relevant information available for each component. The assembled structured representation serves as the input to the subsequent programmatic skill audit, providing a shared basis for systematic analysis. All evidence-bearing inputs to Eq. (3) are strictly restricted to those permitted by the evidence boundary defined above.

3.3 Programmatic Skill Audit

ROSE uses a single LLM call to jointly audit the three IR components. Drawing on interface coverage, contract alignment, and invariant preservation from static program analysis, the audit compares skill procedures with task requirements and artifact properties to identify procedural deficiencies and predictable execution risks. We formulate this stage as:

$$\mathcal{F} = \mathrm{Audit}_\phi(Z), \tag{4}$$

where $\mathcal{F}$ denotes the resulting audit report. The audit prompt requests findings supported by static evidence, corresponding edit recommendations, and identification of useful procedures to preserve. It also instructs the model to distinguish well-supported findings from tentative concerns that should not guide revision. We discuss the resulting findings from three complementary perspectives, corresponding to the audit boxes in Figure 1: *artifact-interface findings*, *operation-contract findings*, and *preservation-and-risk findings*. We detail each category below.

Artifact-interface findings. The audit compares the objects and resources referenced in Z_s with the task requirements and artifact information represented in Z_t and Z_a. Relevant entities include files, tables, output paths, and answer fields. Findings identify missing object references or insufficient guidance on which entities should be inspected, modified, preserved, or produced. These findings guide synthesis to make the corresponding interface requirements explicit.

Operation-contract findings. The audit examines whether the procedures in Z_s cover the required operations, output contracts, and completion conditions specified in Z_t. Findings identify omitted steps or procedural guidance that is too general to establish how a requirement should be met. Examples include an underspecified computation, a missing output-saving step, or an unclear answer format. These findings guide the addition or clarification of procedural instructions.

Preservation-and-risk findings. The audit examines the procedures and invariants in Z_s against the constraints and artifact properties represented in Z_t and Z_a. It identifies useful procedures to retain and predictable execution risks, such as schema mismatches, brittle range assumptions, unsupported tool behavior, or unbounded retry rules. These findings guide synthesis to preserve useful procedures while introducing localized corrections or bounded recovery rules.

3.4 Rollout-Free Static Skill Synthesis

The final stage uses a single LLM call to synthesize an updated skill from the initial skill, the IR, and the audit report:

$$S^* = \mathrm{Synthesize}_\phi(S_0, Z, \mathcal{F}), \tag{5}$$

where S^* denotes the resulting updated skill whose main document is returned in Markdown format. The synthesis prompt requests localized, evidence-supported revisions while preserving useful existing procedures in the initial skill. The corresponding preservation, revision, and deployment requirements illustrated in Figure 1 are described in more detail below.

Preserve procedures. The initial skill S_0 provides the starting point and basis for revision. The synthesis prompt explicitly instructs the model to retain useful existing procedures and invariants, while using the audit report to identify procedural omissions and potential risks that warrant changes. Only the main skill document is revised during this stage; auxiliary resources remain unchanged.

Apply bounded edits. Boundedness is explicitly specified through the synthesis prompt, which requests localized edits supported by the available audit findings. Such edits may add missing task requirements, clarify ambiguous procedural instructions, remove redundant guidance, or introduce bounded recovery rules for foreseeable execution failures.

Emit and freeze the skill. The synthesis call produces a single updated skill without rollout-based candidate evaluation or selection during construction. The updated skill and intermediate construction records are saved for subsequent inspection and reproducibility. The resulting skill is then frozen and reused unchanged across all held-out evaluation tasks:

$$S_x = S^*, \qquad \forall x \in D_{\mathrm{test}}, \tag{6}$$

where S_x denotes the skill supplied to the agent for task x. The evaluation agent, task harness, and evaluator remain fixed within each evaluation setting. ROSE performs no per-task skill rewriting or feedback-driven skill updates during held-out evaluation.

4 Experiments

In this section, we first describe the experimental setup in Section 4.1 and then present and analyze the main results in Section 4.2.

4.1 Experimental Setup

Benchmarks. We evaluate ROSE on four benchmarks covering spreadsheet manipulation, mathematical reasoning, and document question answering. SpreadsheetBench (Ma et al., 2024) requires agents to inspect, modify, and save spreadsheet artifacts. LiveMathematicianBench (He et al., 2026), abbreviated as LiveMath, evaluates mathematical reasoning with prescribed answer formats. DocVQA (Mathew et al., 2021) assesses question answering over document images, requiring agents to interpret textual content and document structure. OfficeQA (Opsahl-Ong et al., 2026) evaluates grounded reasoning over documents containing narrative text and tables. Following SkillOpt (Yang et al., 2026), we adopt fixed train/validation/test splits of $80/40/280$ for SpreadsheetBench, $35/18/124$ for LiveMath, $107/53/374$ for DocVQA, and $50/24/172$ for OfficeQA. For ROSE, skill construction uses only training-visible evidence; the validation split is not used for skill revision or selection. The resulting skill is frozen before evaluation on the test split.

Models. We evaluate ROSE using four agent models in the main experiments: GPT-5.4-Nano (OpenAI, 2026b), GPT-5.4-Mini (OpenAI, 2026b), GPT-5.4 (OpenAI, 2026a), and Qwen3.6-35B-A3B (Qwen Team, 2026). These models span different model families and capacity levels, allowing

Benchmark	Model	No Skill	Human	LLM	Trace2Skill	TextGrad	GEPA	SkillOpt	**ROSE**
Spreadsheet	GPT-5.4	41.4	59.3	46.1	53.2	38.6	61.1	62.5	71.1
	GPT-5.4-Mini	36.1	42.9	36.8	40.7	38.2	42.5	47.5	54.3
	GPT-5.4-Nano	23.5	41.8	33.6	35.4	32.5	37.2	42.5	60.4
	Qwen3.6-35B-A3B	38.2	44.3	42.9	33.2	22.9	45.4	47.5	58.2
	Avg.	34.8	47.1	39.9	40.6	33.1	46.6	50.0	61.0
LiveMath	GPT-5.4	36.8	41.6	36.8	40.0	36.6	41.6	44.0	49.2
	GPT-5.4-Mini	14.7	28.8	28.0	32.8	27.2	27.2	32.8	29.6
	GPT-5.4-Nano	23.2	16.8	20.0	25.6	20.8	24.8	27.2	29.0
	Qwen3.6-35B-A3B	31.2	29.6	24.8	29.6	7.2	31.2	41.6	35.8
	Avg.	26.5	29.2	27.4	32.0	23.0	31.2	36.4	35.9
DocVQA	GPT-5.4	77.6	88.5	90.4	89.3	85.3	88.3	91.2	90.1
	GPT-5.4-Mini	71.4	85.0	86.4	88.5	84.0	83.7	90.9	89.3
	GPT-5.4-Nano	30.8	73.5	71.7	77.3	68.3	71.2	80.2	77.8
	Qwen3.6-35B-A3B	87.6	88.4	88.4	90.4	84.5	88.0	91.4	90.4
	Avg.	66.9	83.9	84.2	86.4	80.5	82.8	88.4	86.9
OfficeQA	GPT-5.4	50.0	59.3	20.4	51.2	54.7	60.4	62.8	56.4
	GPT-5.4-Mini	22.1	45.9	36.6	20.9	30.0	45.3	48.8	46.5
	GPT-5.4-Nano	16.3	46.5	12.8	16.3	38.7	45.0	50.0	50.0
	Qwen3.6-35B-A3B	45.9	41.9	46.5	32.0	33.7	43.6	47.1	47.1
	Avg.	33.6	48.4	29.1	30.1	39.3	48.6	52.2	50.0
Macro Avg.		40.4	52.1	45.1	47.3	44.0	52.3	56.8	58.5

Table 1: Main results across four agent models (GPT-5.4, GPT-5.4-Mini, GPT-5.4-Nano, and Qwen3.6-35B-A3B) on four benchmarks: SpreadsheetBench (Spreadsheet), LiveMathematician-Bench (LiveMath), DocVQA, and OfficeQA. Following SkillOpt (Yang et al., 2026), scores (%) denote hard accuracy for SpreadsheetBench, LiveMath, and OfficeQA, and exact match (EM) for DocVQA; higher is better. Human and LLM denote Human Skill and LLM Skill, respectively. *Avg.* denotes the mean across the four models within each benchmark, and *Macro Avg.* denotes the unweighted mean across all 16 benchmark–model combinations.

us to examine how the effectiveness of skill evolution varies across agents. For each benchmark–model pair, we keep the evaluation model, execution harness, dataset split, and official evaluator fixed across locally evaluated skill conditions. Model parameters remain unchanged, and each constructed skill is frozen throughout held-out evaluation. For ROSE, the same model performs both skill construction and task execution for each benchmark–model pair.

Baselines. We compare ROSE with seven baselines. *No Skill* evaluates the agent with the benchmark's default system prompt and no additional skill document. *Human Skill* provides an expert-written skill document containing benchmark-specific procedural guidance. *LLM Skill* uses a skill generated by an LLM from a high-level task description in a single pass, without subsequent refinement. We also compare against Trace2Skill (Ni et al., 2026), TextGrad (Yuksekgonul et al., 2024), GEPA (Agrawal et al., 2025), and SkillOpt (Yang et al., 2026).

4.2 Main Results

Table 1 presents the main results across four benchmarks and four target models. ROSE achieves competitive held-out performance using only three LLM calls for skill construction, without task rollouts or evaluator feedback. Its macro-average score is 58.5, compared with 56.8 for SkillOpt, 52.3 for GEPA, 47.3 for Trace2Skill, and 44.0 for TextGrad. Together with the construction-cost comparison in Table 3, these results demonstrate that static skill evolution can offer competitive task performance with substantially reduced construction overhead.

ROSE performs particularly well on SpreadsheetBench, achieving an average score of 61.0, compared with 50.0 for SkillOpt, and outperforming all compared methods across the four target models. The largest gain over SkillOpt occurs on GPT-5.4-Nano, with an improvement of 17.9 percentage points (60.4 vs. 42.5). On the remaining three benchmarks, ROSE achieves the second-highest average scores, remaining close to SkillOpt while exceeding Trace2Skill, TextGrad, and GEPA in

Configuration	Acc.
ROSE (Full)	**60.36**
w/o Code-style IR	55.64
w/o Programmatic Skill Audit	51.07
w/o Static Skill Synthesis	33.93
Generic Three-Step Revision†	35.00

Table 2: Ablation and baseline comparison on SpreadsheetBench with GPT-5.4-Nano.

Method	Total tokens	Wall time
ROSE	**9,183**	**32.7 s**
SkillOpt	3,862,634	∼1 h 39 m 52 s
Trace2Skill	4,961,104	∼52 m 52 s

Table 3: Construction-cost comparison using GPT-5.4-Nano and the same 80 SpreadsheetBench training instances.

Skill construction model	SpreadsheetBench				DocVQA			
	Nano	Mini	Grok	GLM	Nano	Mini	Grok	GLM
No Skill	23.5	36.1	40.4	53.2	30.8	71.4	81.3	84.2
GPT-5.4	54.6	57.9	62.1	75.0	75.9	88.2	85.8	89.0
GPT-5.4-Mini	59.6	54.3	61.4	75.7	72.5	89.3	85.3	89.1
GPT-5.4-Nano	60.4	65.0	66.1	77.7	77.8	88.5	89.3	87.6
Qwen3.6-35B-A3B	63.6	67.6	63.6	80.5	75.1	88.8	88.2	91.4

Table 4: Cross-model transfer of ROSE skills on SpreadsheetBench and DocVQA. Skills are constructed using GPT-5.4, GPT-5.4-Mini, GPT-5.4-Nano, or Qwen3.6-35B-A3B, and evaluated using GPT-5.4-Nano (Nano), GPT-5.4-Mini (Mini), Grok-4.6 (Grok), and GLM-5.3-Flash (GLM). Each constructed skill is reused unchanged across evaluation models within the same benchmark. Scores (%; higher is better) follow the evaluation metrics in Table 1.

benchmark-level averages. On LiveMath, it obtains 35.9, compared with 36.4 for SkillOpt, and achieves the best results on GPT-5.4 and GPT-5.4-Nano. On DocVQA and OfficeQA, it reaches 86.9 and 50.0, respectively, compared with SkillOpt's 88.4 and 52.2. These results indicate that rollout-free construction can produce effective skills across the evaluated tasks, with particularly strong benefits for spreadsheet manipulation. The substantial gains on SpreadsheetBench are consistent with the motivation that explicit artifact structures and procedural requirements provide useful evidence for static skill revision.

5 Ablation Study and Further Analysis

Ablation and baseline comparison. Table 2 compares three ablations and a Generic Three-Step Revision baseline on SpreadsheetBench with GPT-5.4-Nano. The full pipeline achieves the highest accuracy of 60.36%. For w/o IR, the initial skill and training-evidence summary are supplied directly to the audit stage, and synthesis uses these inputs together with the resulting audit findings. Accuracy drops to 55.64%, supporting the value of a shared structured representation. For w/o Audit, synthesis uses the initial skill, the full pipeline's IR, and permitted static evidence without audit findings, reducing accuracy to 51.07% and highlighting the benefit of explicit diagnosis. For w/o Synthesis, we re-evaluate the agent with the unchanged initial skill S_0, obtaining 33.93% accuracy. The 26.43-point gap reflects the full pipeline's improvement over the initial skill. Generic Three-Step Revision receives the same initial skill and training evidence as ROSE and uses an LLM to perform three sequential stages of reading, reviewing, and rewriting with generic prompts. It achieves 35.00%, falling 25.36 points below the full pipeline. These results suggest that the proposed structured representation and explicit auditing offer benefits beyond providing the same evidence through a generic three-stage revision process.

Construction cost comparison. Table 3 compares the recorded costs of one complete skill-construction run per method using GPT-5.4-Nano and the same 80 SpreadsheetBench training instances. All measurements come from our own experimental runs and cover the construction phase. ROSE requires only three LLM calls, consuming 9,183 tokens and 32.7 seconds. The native SkillOpt and Trace2Skill pipelines consume 3,862,634 and 4,961,104 tokens, with wall times of ap-

Benchmark	Metric	Run 1	Run 2	Run 3	Mean ± SD
SpreadsheetBench	Accuracy	60.36	58.21	61.79	60.12 ± 1.80
LiveMath	Accuracy	29.03	24.19	26.61	26.61 ± 2.42
DocVQA	Exact Match	73.80	71.39	77.81	74.33 ± 3.24
OfficeQA	Accuracy	50.00	49.42	45.93	48.45 ± 2.20

Table 5: Performance stability of ROSE with GPT-5.4-Nano. Each of three independent runs constructs a new skill and evaluates it on the held-out test split. We report individual scores (%) and their mean ± sample standard deviation (SD, in percentage points). Variability reflects both skill construction and evaluation.

proximately 99 minutes 52 seconds and 52 minutes 52 seconds, respectively. Trace2Skill's elapsed time is estimated from its wrapper start timestamp and final log timestamp. Under these conditions, ROSE reduces total token usage by approximately 99.76% relative to SkillOpt and 99.81% relative to Trace2Skill, with corresponding wall-clock speedups of approximately $183\times$ and $97\times$. These substantial savings arise from its fixed three-stage construction pipeline, which derives skill revisions from static evidence without task rollouts, feedback collection, or repeated candidate evaluation. Together with the performance results in Table 1, this comparison supports the practical value of rollout-free skill evolution: competitive held-out performance can be achieved with substantially lower skill-construction cost in the evaluated setting.

Cross-model skill transfer. To assess the generalizability of ROSE, we conduct skill-transfer experiments on SpreadsheetBench and DocVQA. We use GPT-5.4, GPT-5.4-Mini, GPT-5.4-Nano, and Qwen3.6-35B-A3B as skill construction models. The test models are GPT-5.4-Nano, GPT-5.4-Mini, Grok-4.6, and GLM-5.3-Flash. Beyond the two GPT variants, Grok-4.6 and GLM-5.3-Flash represent two substantially different model families outside those used for construction, allowing us to examine transfer across heterogeneous models. Each constructed skill is frozen and reused unchanged by all four test models within the same benchmark. As shown in Table 4, all evaluated construction–test combinations outperform the corresponding no-skill setting. On SpreadsheetBench, skills produced by the four construction models improve Grok-4.6 from 40.4 to between 61.4 and 66.1, and GLM-5.3-Flash from 53.2 to between 75.0 and 80.5. On DocVQA, the corresponding scores increase from 81.3 to 85.3–89.3 for Grok-4.6 and from 84.2 to 87.6–91.4 for GLM-5.3-Flash. These improvements occur across all four construction models, showing that successful transfer is not limited to a single source model. The most effective constructor nevertheless varies: GPT-5.4-Nano produces the best transferred skills for Grok-4.6 on both benchmarks, whereas Qwen3.6-35B-A3B performs best for GLM-5.3-Flash. These results support the reuse of statically constructed skills across distinct model families without additional skill construction or revision for each test model.

Performance stability. To examine whether the benefits of ROSE persist across independent runs, we repeat the complete skill-construction and evaluation pipeline three times with GPT-5.4-Nano. This experiment is particularly relevant because ROSE produces a single frozen skill without rollout-based candidate evaluation or feedback-guided selection. Each run independently constructs a new skill and evaluates it on the same held-out test split. As reported in Table 5, the mean scores and sample standard deviations are 60.12 ± 1.80 on SpreadsheetBench, 26.61 ± 2.42 on LiveMath, 74.33 ± 3.24 on DocVQA, and 48.45 ± 2.20 on OfficeQA. On SpreadsheetBench, all three independent runs achieve scores between 58.21 and 61.79, consistently exceeding the reported SkillOpt result of 42.5 for the same target model. Thus, the substantial performance advantage observed on this benchmark recurs across independently constructed skills. Every run also exceeds the corresponding no-skill result on all four evaluated benchmarks. The observed standard deviations range from 1.80 to 3.24 percentage points, with the largest variation occurring on DocVQA. These measurements capture the combined variability of both skill construction and task execution, providing further evidence of repeatable improvements under the evaluated setting.

Generalization across model families. To assess whether the effectiveness of ROSE extends beyond the models included in the main comparison, we evaluate four additional models from distinct model families—GPT-5.6-Luna, Gemini 2.5 Flash-Lite, Grok-4.6, and GLM-5.3-Flash—on the same four benchmarks. Each model performs both skill construction and task execution, with the

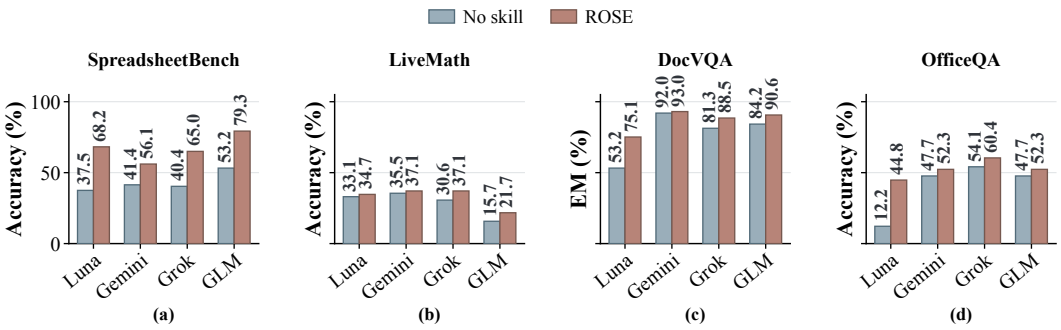

Figure 2: Comparison of ROSE and No Skill across four additional agent models: GPT-5.6-Luna (Luna), Gemini 2.5 Flash-Lite (Gemini), Grok-4.6 (Grok), and GLM-5.3-Flash (GLM). Panels (a)–(d) show results on SpreadsheetBench, LiveMathematicianBench (LiveMath), DocVQA, and OfficeQA, respectively. For ROSE, the same model performs both skill construction and task execution. All scores are reported as percentages.

resulting skill remaining frozen throughout held-out evaluation. As shown in Figure 2, ROSE outperforms No Skill across all 16 evaluated benchmark–model combinations. On SpreadsheetBench, gains range from 14.7 to 30.7 percentage points, with GPT-5.6-Luna improving from 37.5 to 68.2 and GLM-5.3-Flash from 53.2 to 79.3. These consistent gains across all four models further support the effectiveness of static skill evolution for spreadsheet manipulation. Improvements also extend to the other three benchmarks. On LiveMath, gains range from 1.6 to 6.5 points. On DocVQA, GPT-5.6-Luna improves from 53.2 to 75.1 and Grok-4.6 from 81.3 to 88.5, while Gemini 2.5 Flash-Lite and GLM-5.3-Flash also improve despite their higher baseline scores. On OfficeQA, GPT-5.6-Luna improves from 12.2 to 44.8, and the remaining models gain between 4.6 and 6.3 points. Although the magnitude of improvement varies across models and benchmarks, these consistently positive results provide further support for the applicability of the same rollout-free construction pipeline across the evaluated model families, without relying on execution feedback during skill construction.

6 Limitations

First, our evaluation focuses on single-agent settings, leaving the effectiveness of ROSE in multi-agent systems, where communication and coordination introduce additional requirements, unexplored. Second, ROSE primarily addresses procedural deficiencies that can be inferred from static task evidence. Its ability to address missing domain knowledge, failures that emerge only during execution, and recovery strategies requiring multi-step interaction remains limited. Third, ROSE constructs a single frozen benchmark-level skill using predefined audit categories; modular skill composition and automatic discovery of additional risk patterns remain directions for future work. Finally, our implementation focuses on offline skill construction, leaving incremental updates and ongoing maintenance of evolving skill repositories unexplored.

7 Conclusion

We introduced ROSE, a rollout-free framework for efficient agent skill evolution inspired by static program analysis. ROSE organizes the initial skill, training-task specifications, and visible artifact metadata into a shared code-style intermediate representation, audits it for procedural omissions, contract mismatches, and predictable execution risks, and uses the findings to guide localized skill revisions while preserving useful procedures. Construction requires only three LLM calls, and the resulting skill remains unchanged throughout held-out evaluation. Across four benchmarks, ROSE achieves competitive performance, with particularly strong gains on spreadsheet manipulation. A matched SpreadsheetBench comparison demonstrates substantial reductions in token usage and construction time. Additional experiments support its effectiveness across diverse model families and the transferability of the constructed skills. These findings highlight the potential of static task evidence to support effective, reusable skill updates at low construction cost, providing a practical foundation for further research on rollout-free agent skill evolution.

AI USE STATEMENT

We used generative AI tools both as a component of the method under study and as research assistants. In ROSE, language models perform code-style IR construction, skill auditing, and skill synthesis. Each construction run produces a reusable skill through three sequential LLM calls, which is then frozen for downstream evaluation. These model-generated artifacts are part of the experimental method rather than independently verified instructions or formal correctness guarantees. We also used AI assistants to support code implementation, debugging, experiment orchestration, prompt and benchmark-adapter development, analysis of experimental outputs, and manuscript refinement. The authors are responsible for verifying the AI-assisted code, experimental records, interpretations, and manuscript, and take responsibility for the final content and claims of this work.

ETHICS STATEMENT

ROSE aims to improve the reliability and efficiency of language-model agents without updating model weights. Potential benefits include more reliable spreadsheet manipulation, mathematical reasoning, and document question answering. However, generated skills may contain unsupported assumptions or propagate errors across multiple tasks. Improved agent capabilities may also be misused, and performance on the evaluated benchmarks does not establish safety for deployment in high-stakes settings.

We recommend reviewing generated skills before deployment, restricting tool permissions, and retaining human oversight for consequential actions. Dataset use and artifact redistribution should respect the original licenses and access conditions. When using external model APIs, researchers should consider the privacy implications of transmitting task content and should exclude credentials and sensitive information from released prompts, logs, and other artifacts.

REPRODUCIBILITY STATEMENT

Section 3 describes ROSE's three-stage construction procedure and evidence boundary. Section 4.1 specifies the benchmarks, dataset split sizes, model settings, and evaluation protocol, while Section 5 presents ablations, construction-cost measurements, cross-model transfer, and repeated-run analyses. The appendix provides a complete construction example, including the prompts, intermediate records, and resulting skill.

We will release the implementation, benchmark-specific evidence preparation code, initial and final skills, construction prompts and responses, split identifiers, evaluation configurations, and experimental results, subject to dataset licenses and access restrictions. The release will document the model and provider configurations and any preprocessing or recovery procedures needed to interpret the reported runs. API credentials will not be included; restricted benchmark assets will be accompanied by acquisition instructions rather than redistributed.

Exact numerical reproduction may be affected by stochastic generation, model-service updates, and provider-specific behavior. Releasing frozen skills and construction records will support separate reproduction of downstream evaluation and inspection of the skill-construction process.

REFERENCES

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, et al. Gepa: Reflective prompt evolution can outperform reinforcement learning. *arXiv preprint arXiv:2507.19457*, 2025.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Shiqi Chen, Jingze Gai, Ruochen Zhou, Jinghan Zhang, Tongyao Zhu, Junlong Li, Kangrui Wang, Zihan Wang, Zhengyu Chen, Klara Kaleb, Ning Miao, Siyang Gao, Cong Lu, Manling Li, Junxian

He, and Yee Whye Teh. Skillcraft: Can llm agents learn to use tools skillfully? *arXiv preprint arXiv:2603.00718*, 2026.

Linyang He, Qiyao Yu, Hanze Dong, Baohao Liao, Xinxing Xu, Micah Goldblum, Jiang Bian, and Nima Mesgarani. Livemathematicianbench: A live benchmark for mathematician-level reasoning with proof sketches. *arXiv preprint arXiv:2604.01754*, 2026.

Yanna Jiang, Delong Li, Haiyu Deng, Baihe Ma, Xu Wang, Qin Wang, and Guangsheng Yu. Sok: Agentic skills – beyond tool use in llm agents. *arXiv preprint arXiv:2602.20867*, 2026.

Hao Li, Chunjiang Mu, Jianhao Chen, Siyue Ren, Zhiyao Cui, Yiqun Zhang, Lei Bai, and Shuyue Hu. Organizing, orchestrating, and benchmarking agent skills at ecosystem scale. *arXiv preprint arXiv:2603.02176*, 2026a.

Xinze Li, Yuhang Zang, Yixin Cao, and Aixin Sun. Skill-as-pseudocode: Refactoring skill libraries to pseudocode for llm agents. *arXiv preprint arXiv:2605.27955*, 2026b.

George Ling, Shanshan Zhong, and Richard Huang. Agent skills: A data-driven analysis of claude skills for extending large language model functionality. *arXiv preprint arXiv:2602.08004*, 2026.

Hongjun Liu, Yifei Ming, Shafiq Joty, and Chen Zhao. Harnessing llm agents with skill programs. *arXiv preprint arXiv:2605.17734*, 2026.

Zeyao Ma, Bohan Zhang, Jing Zhang, Jifan Yu, Xiaokang Zhang, Xiaohan Zhang, Sijia Luo, Xi Wang, and Jie Tang. Spreadsheetbench: Towards challenging real world spreadsheet manipulation. *Advances in Neural Information Processing Systems*, 37:94871–94908, 2024.

Ziyu Ma, Shidong Yang, Yuxiang Ji, Xucong Wang, Yong Wang, Yiming Hu, Tongwen Huang, and Xiangxiang Chu. Skillclaw: Let skills evolve collectively with agentic evolver. *arXiv preprint arXiv:2604.08377*, 2026.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2199–2208. IEEE, 2021.

Jingwei Ni, Yihao Liu, Xinpeng Liu, Yutao Sun, Mengyu Zhou, Pengyu Cheng, Dexin Wang, Erchao Zhao, Xiaoxi Jiang, and Guanjun Jiang. Trace2skill: Distill trajectory-local lessons into transferable agent skills. *arXiv preprint arXiv:2603.25158*, 2026.

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

OpenAI. Introducing GPT-5.4 Mini and Nano, March 2026b. URL https://openai.com/index/introducing-gpt-5-4-mini-and-nano/.

Krista Opsahl-Ong, Arnav Singhvi, Jasmine Collins, Ivan Zhou, Cindy Wang, Ashutosh Baheti, Owen Oertell, Jacob Portes, Sam Havens, Erich Elsen, et al. Officeqa pro: An enterprise benchmark for end-to-end grounded reasoning. *arXiv preprint arXiv:2603.08655*, 2026.

Qianjun Pan, Yutao Yang, Junsong Li, Jie Zhou, Kai Chen, Xin Li, Qin Chen, and Liang He. Anything2skill: Compiling external knowledge into reusable skills for agents. *arXiv preprint arXiv:2606.09316*, 2026.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all, April 2026. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 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.

Yu Tian, Jiawei Chen, Lifan Zheng, Mingxiang Tao, Xinyi Zeng, Zhaoxia Yin, Hang Su, and Xian Sun. Skills-coach: A self-evolving skill optimizer via training-free grpo. *arXiv preprint arXiv:2604.27488*, 2026.

Yash Vishe, Rohan Surana, Xunyi Jiang, Zihan Huang, Xintong Li, Nikki Lijing Kuang, Tong Yu, Ryan A Rossi, Jingbo Shang, Julian McAuley, et al. Skill-r1: Agent skill evolution via reinforcement learning. *arXiv preprint arXiv:2605.09359*, 2026.

Hanyu Wang, Yifan Lan, Bochuan Cao, Lu Lin, and Jinghui Chen. Skillgrad: Optimizing agent skills like gradient descent. *arXiv preprint arXiv:2605.27760*, 2026.

Renjun Xu and Yang Yan. Agent skills for large language models: Architecture, acquisition, security, and the path forward. *arXiv preprint arXiv:2602.12430*, 2026.

John Yang, Carlos Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652, 2024.

Yifan Yang, Ziyang Gong, Weiquan Huang, Qihao Yang, Ziwei Zhou, Zisu Huang, Yan Li, Xuemei Gao, Qi Dai, Bei Liu, et al. Skillopt: Executive strategy for self-evolving agent skills. *arXiv preprint arXiv:2605.23904*, 2026.

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*, 2022.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. Textgrad: Automatic" differentiation" via text. *arXiv preprint arXiv:2406.07496*, 2024.

Chenyu Zhou, Huacan Chai, Wenteng Chen, Zihan Guo, Rong Shan, Yuanyi Song, Tianyi Xu, Yingxuan Yang, Aofan Yu, Weiming Zhang, Congming Zheng, Jiachen Zhu, Zeyu Zheng, Zhuosheng Zhang, Xingyu Lou, Changwang Zhang, Zhihui Fu, Jun Wang, Weiwen Liu, Jianghao Lin, and Weinan Zhang. Externalization in llm agents: A unified review of memory, skills, protocols and harness engineering. *arXiv preprint arXiv:2604.08224*, 2026.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, volume 2024, pp. 15585–15606, 2024.

8 A COMPLETE DOCVQA SKILL-CONSTRUCTION EXAMPLE

Initial skill

```
# DocVQA Skill

## Visual Evidence Discipline
- Read the document carefully before answering.
- Prefer the smallest exact text span that answers the question.
- When several nearby strings look similar, choose the one whose
↪  surrounding labels or layout best match the question.

## Exact Answer Discipline
- Copy names, numbers, and dates exactly from the document whenever
↪  possible.
- Prefer direct extraction over paraphrase.
- Before finalizing, compare the answer against nearby alternatives
↪  and keep the best-supported exact span.
```

Training evidence summary


```
{
  "benchmark": "docvqa",
  "task_specifications": {
    "train_item_count": 107,
    "operation_family_counts": {
      "comparison": 3,
      "count": 8,
      "date_time": 10,
      "direct_lookup": 43,
      "document_type": 1,
      "entity_name": 25,
      "field_binding": 8,
      "location": 7,
      "quantity": 12,
      "table_lookup": 5,
      "yes_no": 1
    },
    "document_topic_counts": {
      "Image/Photo": 1,
      "Yes/No|handwritten|form": 1,
      "figure/diagram": 2,
      "figure/diagram|layout": 1,
      "form": 14,
      "form|layout": 2,
      "form|table/list": 1,
      "free_text": 12,
      "free_text|layout": 4,
      "free_text|others": 1,
      "handwritten|form": 6,
      "handwritten|table/list": 2,
      "layout": 24,
      "layout|Image/Photo": 1,
      "layout|others": 2,
      "others": 2,
      "table/list": 29,
      "table/list|layout": 1,
      "table/list|others": 1
    },
    "required_procedures_for_observed_train_categories": {
      "question_conditioned": {},
      "layout_conditioned": {}
```

```
  },
  "representative_examples_without_answers_or_image_text": [
    {
      "id": "44841",
      "question": "Which product shows a higher percentage of
      ↪  'extremely ' likely usage when compared to 'very ' likely
      ↪  usage;  UXL or Marathon ?",
      "operation_tags": [
        "quantity",
        "date_time",
        "comparison"
      ],
      "document_topic": "figure/diagram"
    },
    {
      "id": "12577",
      "question": "What is the location of the fax receiver ?",
      "operation_tags": [
        "field_binding",
        "location"
      ],
      "document_topic": "form"
    },
    {
      "id": "57145",
      "question": "How many points are there in section II of the
      ↪  second table?",
      "operation_tags": [
        "count",
        "table_lookup"
      ],
      "document_topic": "table/list|others"
    },
    {
      "id": "1182",
      "question": "in which year agency will come out with
      ↪  guidelines ?",
      "operation_tags": [
        "date_time",
        "entity_name"
      ],
      "document_topic": "free_text"
    },
    {
      "id": "17001",
      "question": "What type of form is this?",
      "operation_tags": [
        "field_binding",
        "document_type"
      ],
      "document_topic": "layout"
    },
    {
      "id": "42304",
      "question": "what is the sales in 2013?",
      "operation_tags": [
        "direct_lookup"
      ],
      "document_topic": "table/list"
    },
    {
      "id": "56464",
      "question": "Did Michael have any graduate level course in
      ↪  epidemiology?",
      "operation_tags": [
```

```
        "yes_no"
      ],
      "document_topic": "Yes/No|handwritten|form"
    },
    {
      "id": "17150",
      "question": "What is the Exhibit No mentioned in the
      ↪  document?",
      "operation_tags": [
        "field_binding"
      ],
      "document_topic": "others"
    },
    {
      "id": "271",
      "question": "What date is written on this document?",
      "operation_tags": [
        "date_time"
      ],
      "document_topic": "handwritten|form"
    },
    {
      "id": "51244",
      "question": "What is the middle name of John?",
      "operation_tags": [
        "entity_name"
      ],
      "document_topic": "form|table/list"
    },
    {
      "id": "5388",
      "question": "what is the carbon intensity of wood according
      ↪  to graph ?",
      "operation_tags": [
        "direct_lookup"
      ],
      "document_topic": "figure/diagram|layout"
    },
    {
      "id": "5612",
      "question": "Bale at various moisture contents are for how
      ↪  many stalk positions?",
      "operation_tags": [
        "count"
      ],
      "document_topic": "free_text|layout"
    },
    {
      "id": "57087",
      "question": "What is obtained by deducting  depreciation from
      ↪  'Gross Block' (gross block - depreciation)?",
      "operation_tags": [
        "direct_lookup"
      ],
      "document_topic": "table/list|layout"
    },
    {
      "id": "5778",
      "question": "What is the name of the city mentioned?",
      "operation_tags": [
        "entity_name",
        "location"
      ],
      "document_topic": "form|layout"
    },
```

```
      {
        "id": "59304",
        "question": "What is the name of the company?",
        "operation_tags": [
          "entity_name"
        ],
        "document_topic": "layout|Image/Photo"
      },
      {
        "id": "59796",
        "question": "What is the height and width of the Met
        ↪  Mirror?",
        "operation_tags": [
          "direct_lookup"
        ],
        "document_topic": "free_text|others"
      }
    ]
  },
  "artifact_metadata": {
    "artifact_type": "single document-page image",
    "visible_fields": [
      "question",
      "document image"
    ],
    "analysis_contract": "aggregate train-only pixel geometry; no
    ↪  OCR, recognized text, or answers",
    "layout_profile_counts": {
      "dense_text_or_graphics": 3,
      "horizontally_ruled": 20,
      "mixed_layout": 13,
      "multi_column": 7,
      "ruled_grid": 58,
      "sparse_form_or_note": 6
    },
    "image_width": {
      "min": 593,
      "median": 1702,
      "p90": 2207,
      "max": 5325
    },
    "image_height": {
      "min": 294,
      "median": 2214,
      "p90": 2319,
      "max": 7150
    },
    "image_pixel_count_thousands": {
      "min": 174,
      "median": 3800,
      "p90": 4138,
      "max": 38010
    },
    "contrast_span": {
      "min": 0,
      "median": 159,
      "p90": 255,
      "max": 255
    },
    "foreground_density_basis_points": {
      "min": 299,
      "median": 1155,
      "p90": 2349,
      "max": 9863
    },
```

```
    "ink_bbox_coverage_basis_points": {
      "min": 5166,
      "median": 9827,
      "p90": 10000,
      "max": 10000
    },
    "occupied_row_bands": {
      "min": 1,
      "median": 4,
      "p90": 31,
      "max": 59
    },
    "long_horizontal_rules": {
      "min": 0,
      "median": 2,
      "p90": 11,
      "max": 37
    },
    "long_vertical_rules": {
      "min": 0,
      "median": 2,
      "p90": 5,
      "max": 31
    },
    "whitespace_column_gaps": {
      "min": 0,
      "median": 0,
      "p90": 2,
      "max": 3
    },
    "static_hazards": []
  },
  "public_contracts": {
    "semantics": "Answer the question from the attached DocVQA
    ↪  document-page image.",
    "operations": "perform the requested lookup, field binding,
    ↪  comparison, count, quantity, date/time, entity, location, or
    ↪  document-type operation using visible evidence",
    "output": "return one concise answer inside
    ↪  <answer>...</answer>",
    "completion": "one image-grounded answer expression is emitted"
  },
  "excluded_fields": [
    "answer",
    "answers",
    "ocr_text",
    "recognized_image_text",
    "validation_content",
    "test_content"
  ]
}
```


IR construction: complete prompt

```
You are constructing a code-style intermediate representation (IR)
↪  for rollout-free static skill evolution.

Benchmark: docvqa

Goal: convert the initial skill and train-visible static evidence
↪  into a typed IR record.

Allowed evidence:
```

````
- The initial skill text below.
- The train-visible evidence summary below.
- Public benchmark semantics below.

Forbidden evidence:
- Held-out/test labels or rewards.
- Hidden answers, correct labels, reference solutions, or verifier
↪  outcomes.
- Held-out trajectories or per-test failure traces.
- Per-test rewriting.

Output a concise typed IR record with exactly these sections:
1. Skill IR: tool_interface, procedure, preservation_rules,
↪  recovery_rules.
2. Task Contract IR: intent, operations, output_contract,
↪  completion_requirement.
3. Artifact IR: artifact_type, structure, locations, hazards.

Rules:
- Extract only facts explicitly stated or directly implied by the
↪  allowed evidence.
- If evidence for a field is weak, write "unspecified" rather than
↪  inventing.
- Do not output an updated skill.
- Output only the IR record in Markdown.

Public benchmark semantics:
Answer the question from the attached DocVQA document-page image.

Initial skill:
```markdown
# DocVQA Skill

## Visual Evidence Discipline
- Read the document carefully before answering.
- Prefer the smallest exact text span that answers the question.
- When several nearby strings look similar, choose the one whose
↪  surrounding labels or layout best match the question.

## Exact Answer Discipline
- Copy names, numbers, and dates exactly from the document whenever
↪  possible.
- Prefer direct extraction over paraphrase.
- Before finalizing, compare the answer against nearby alternatives
↪  and keep the best-supported exact span.

```

Train-visible evidence summary:
```json
{
  "benchmark": "docvqa",
  "task_specifications": {
    "train_item_count": 107,
    "operation_family_counts": {
      "comparison": 3,
      "count": 8,
      "date_time": 10,
      "direct_lookup": 43,
      "document_type": 1,
      "entity_name": 25,
      "field_binding": 8,
      "location": 7,
      "quantity": 12,
````
````

```
      "table_lookup": 5,
      "yes_no": 1
    },
    "document_topic_counts": {
      "Image/Photo": 1,
      "Yes/No|handwritten|form": 1,
      "figure/diagram": 2,
      "figure/diagram|layout": 1,
      "form": 14,
      "form|layout": 2,
      "form|table/list": 1,
      "free_text": 12,
      "free_text|layout": 4,
      "free_text|others": 1,
      "handwritten|form": 6,
      "handwritten|table/list": 2,
      "layout": 24,
      "layout|Image/Photo": 1,
      "layout|others": 2,
      "others": 2,
      "table/list": 29,
      "table/list|layout": 1,
      "table/list|others": 1
    },
    "required_procedures_for_observed_train_categories": {
      "question_conditioned": {},
      "layout_conditioned": {}
    },
    "representative_examples_without_answers_or_image_text": [
      {
        "id": "44841",
        "question": "Which product shows a higher percentage of
        ↪  'extremely ' likely usage when compared to 'very ' likely
        ↪  usage;  UXL or Marathon ?",
        "operation_tags": [
          "quantity",
          "date_time",
          "comparison"
        ],
        "document_topic": "figure/diagram"
      },
      {
        "id": "12577",
        "question": "What is the location of the fax receiver ?",
        "operation_tags": [
          "field_binding",
          "location"
        ],
        "document_topic": "form"
      },
      {
        "id": "57145",
        "question": "How many points are there in section II of the
        ↪  second table?",
        "operation_tags": [
          "count",
          "table_lookup"
        ],
        "document_topic": "table/list|others"
      },
      {
        "id": "1182",
        "question": "in which year agency will come out with
        ↪  guidelines ?",
        "operation_tags": [
```

```
        "date_time",
        "entity_name"
      ],
      "document_topic": "free_text"
    },
    {
      "id": "17001",
      "question": "What type of form is this?",
      "operation_tags": [
        "field_binding",
        "document_type"
      ],
      "document_topic": "layout"
    },
    {
      "id": "42304",
      "question": "what is the sales in 2013?",
      "operation_tags": [
        "direct_lookup"
      ],
      "document_topic": "table/list"
    },
    {
      "id": "56464",
      "question": "Did Michael have any graduate level course in
      ↪  epidemiology?",
      "operation_tags": [
        "yes_no"
      ],
      "document_topic": "Yes/No|handwritten|form"
    },
    {
      "id": "17150",
      "question": "What is the Exhibit No mentioned in the
      ↪  document?",
      "operation_tags": [
        "field_binding"
      ],
      "document_topic": "others"
    },
    {
      "id": "271",
      "question": "What date is written on this document?",
      "operation_tags": [
        "date_time"
      ],
      "document_topic": "handwritten|form"
    },
    {
      "id": "51244",
      "question": "What is the middle name of John?",
      "operation_tags": [
        "entity_name"
      ],
      "document_topic": "form|table/list"
    },
    {
      "id": "5388",
      "question": "what is the carbon intensity of wood according
      ↪  to graph ?",
      "operation_tags": [
        "direct_lookup"
      ],
      "document_topic": "figure/diagram|layout"
    },
```

```
      {
        "id": "5612",
        "question": "Bale at various moisture contents are for how
        ↪  many stalk positions?",
        "operation_tags": [
          "count"
        ],
        "document_topic": "free_text|layout"
      },
      {
        "id": "57087",
        "question": "What is obtained by deducting  depreciation from
        ↪  'Gross Block' (gross block - depreciation)?",
        "operation_tags": [
          "direct_lookup"
        ],
        "document_topic": "table/list|layout"
      },
      {
        "id": "5778",
        "question": "What is the name of the city mentioned?",
        "operation_tags": [
          "entity_name",
          "location"
        ],
        "document_topic": "form|layout"
      },
      {
        "id": "59304",
        "question": "What is the name of the company?",
        "operation_tags": [
          "entity_name"
        ],
        "document_topic": "layout|Image/Photo"
      },
      {
        "id": "59796",
        "question": "What is the height and width of the Met
        ↪  Mirror?",
        "operation_tags": [
          "direct_lookup"
        ],
        "document_topic": "free_text|others"
      }
    ]
  },
  "artifact_metadata": {
    "artifact_type": "single document-page image",
    "visible_fields": [
      "question",
      "document image"
    ],
    "analysis_contract": "aggregate train-only pixel geometry; no
    ↪  OCR, recognized text, or answers",
    "layout_profile_counts": {
      "dense_text_or_graphics": 3,
      "horizontally_ruled": 20,
      "mixed_layout": 13,
      "multi_column": 7,
      "ruled_grid": 58,
      "sparse_form_or_note": 6
    },
    "image_width": {
      "min": 593,
      "median": 1702,
```

```
        "p90": 2207,
        "max": 5325
      },
      "image_height": {
        "min": 294,
        "median": 2214,
        "p90": 2319,
        "max": 7150
      },
      "image_pixel_count_thousands": {
        "min": 174,
        "median": 3800,
        "p90": 4138,
        "max": 38010
      },
      "contrast_span": {
        "min": 0,
        "median": 159,
        "p90": 255,
        "max": 255
      },
      "foreground_density_basis_points": {
        "min": 299,
        "median": 1155,
        "p90": 2349,
        "max": 9863
      },
      "ink_bbox_coverage_basis_points": {
        "min": 5166,
        "median": 9827,
        "p90": 10000,
        "max": 10000
      },
      "occupied_row_bands": {
        "min": 1,
        "median": 4,
        "p90": 31,
        "max": 59
      },
      "long_horizontal_rules": {
        "min": 0,
        "median": 2,
        "p90": 11,
        "max": 37
      },
      "long_vertical_rules": {
        "min": 0,
        "median": 2,
        "p90": 5,
        "max": 31
      },
      "whitespace_column_gaps": {
        "min": 0,
        "median": 0,
        "p90": 2,
        "max": 3
      },
      "static_hazards": []
    },
    "public_contracts": {
      "semantics": "Answer the question from the attached DocVQA
      ↪  document-page image.",
      "operations": "perform the requested lookup, field binding,
      ↪  comparison, count, quantity, date/time, entity, location, or
      ↪  document-type operation using visible evidence",
```

```
    "output": "return one concise answer inside
    ↪  <answer>...</answer>",
    "completion": "one image-grounded answer expression is emitted"
  },
  "excluded_fields": [
    "answer",
    "answers",
    "ocr_text",
    "recognized_image_text",
    "validation_content",
    "test_content"
  ]
}
```

IR construction: complete response

```
## 1. Skill IR
- **tool_interface:** `input = {question, document-page image}`;
↪  `output = one concise answer string`
- **procedure:**
  1. Inspect the document page carefully.
  2. Use visible layout/text evidence to identify the answer target.
  3. Prefer the smallest exact text span that answers the question.
  4. For nearby similar strings, select the one best supported by
  ↪  surrounding labels/layout.
  5. Copy names, numbers, and dates exactly when possible.
- **preservation_rules:**
  - Keep answers grounded in the visible document image.
  - Prefer direct extraction over paraphrase.
  - Preserve exact spelling, punctuation, and formatting when
  ↪  extracting spans.
  - Compare nearby alternatives before finalizing.
- **recovery_rules:**
  - If multiple candidate spans are similar, fall back to the span
  ↪  with the strongest local layout/label support.
  - If the answer cannot be localized confidently, return the
  ↪  best-supported minimal exact span; otherwise unspecified.

## 2. Task Contract IR
- **intent:** answer the question from the attached DocVQA
↪  document-page image.
- **operations:** visible-evidence lookup; field binding; comparison;
↪  count; quantity extraction; date/time extraction; entity name
↪  extraction; location extraction; document-type identification.
- **output_contract:** emit exactly one concise answer inside
↪  `<answer>...</answer>`.
- **completion_requirement:** one image-grounded answer expression is
↪  emitted.

## 3. Artifact IR
- **artifact_type:** single document-page image
- **structure:** page-level document image paired with a question;
↪  observed layouts include ruled grids, horizontally ruled pages,
↪  multi-column pages, mixed layouts, sparse forms/notes, dense
↪  text/graphics, free text, tables/lists, handwritten forms, and
↪  diagrams.
- **locations:** question text is provided separately from the
↪  document image; evidence is located within the visible page
↪  image.
```

```
- **hazards:** layout ambiguity; visually similar nearby strings;
↪  dense tables/grids; mixed handwriting/printed content; no OCR or
↪  recognized text is guaranteed.
```

Joint audit: complete prompt

````
You are performing Programmatic Skill Audit for rollout-free static
↪  skill evolution.

Benchmark: docvqa

Goal: identify high-confidence contract mismatches and static risks
↪  from the code-style IR below.

Allowed evidence:
- The IR record below, constructed from the initial skill and
↪  train-visible static evidence.
- Public benchmark semantics already recorded in the IR.

Forbidden evidence:
- Held-out/test labels or rewards.
- Hidden answers, correct labels, reference solutions, or verifier
↪  outcomes.
- Held-out trajectories or per-test failure traces.
- Per-test rewriting.

Output a concise audit report with:
1. detected static signals,
2. contract mismatch findings,
3. preservation risks,
4. supported bounded edit recommendations,
5. findings that should NOT drive edits because evidence is weak.

Do not output an updated skill. Output only the audit report.

Code-style IR record:
```markdown
## 1. Skill IR
- **tool_interface:** `input = {question, document-page image}`;
↪  `output = one concise answer string`
- **procedure:**
  1. Inspect the document page carefully.
  2. Use visible layout/text evidence to identify the answer target.
  3. Prefer the smallest exact text span that answers the question.
  4. For nearby similar strings, select the one best supported by
  ↪  surrounding labels/layout.
  5. Copy names, numbers, and dates exactly when possible.
- **preservation_rules:**
  - Keep answers grounded in the visible document image.
  - Prefer direct extraction over paraphrase.
  - Preserve exact spelling, punctuation, and formatting when
  ↪  extracting spans.
  - Compare nearby alternatives before finalizing.
- **recovery_rules:**
  - If multiple candidate spans are similar, fall back to the span
  ↪  with the strongest local layout/label support.
  - If the answer cannot be localized confidently, return the
  ↪  best-supported minimal exact span; otherwise unspecified.

## 2. Task Contract IR
- **intent:** answer the question from the attached DocVQA
↪  document-page image.
````
````

```
- **operations:** visible-evidence lookup; field binding; comparison;
↪  count; quantity extraction; date/time extraction; entity name
↪  extraction; location extraction; document-type identification.
- **output_contract:** emit exactly one concise answer inside
↪  `<answer>...</answer>`.
- **completion_requirement:** one image-grounded answer expression is
↪  emitted.

## 3. Artifact IR
- **artifact_type:** single document-page image
- **structure:** page-level document image paired with a question;
↪  observed layouts include ruled grids, horizontally ruled pages,
↪  multi-column pages, mixed layouts, sparse forms/notes, dense
↪  text/graphics, free text, tables/lists, handwritten forms, and
↪  diagrams.
- **locations:** question text is provided separately from the
↪  document image; evidence is located within the visible page
↪  image.
- **hazards:** layout ambiguity; visually similar nearby strings;
↪  dense tables/grids; mixed handwriting/printed content; no OCR or
↪  recognized text is guaranteed.
```

Joint audit: complete response

```
## 1. Detected static signals
- **Output formatting conflict risk:** Skill IR says `output = one
↪  concise answer string`, while Task Contract IR requires
↪  **wrapping the answer in `<answer>...</answer>`**.
- **Always-output constraint tension:** Skill "recovery_rules"
↪  includes: "**If the answer cannot be localized confidently...;
↪  otherwise unspecified.**" This can yield **empty/undefined
↪  output**, conflicting with "completion_requirement: one
↪  image-grounded answer expression is emitted."
- **Grounding vs. evaluation normalization uncertainty:** Skill
↪  preservation rules demand **exact
↪  spelling/punctuation/formatting**. DocVQA scoring often tolerates
↪  some formatting variance via normalization (exact details vary),
↪  so "exact span only" can become a **precision risk** if
↪  normalization is expected.
- **No OCR/recognition guaranteed (artifact hazard):** IR explicitly
↪  notes **no OCR/recognized text is guaranteed**, yet the procedure
↪  strongly relies on "smallest exact text span" extraction and
↪  "copy names/numbers/dates exactly."

## 2. Contract mismatch findings (high-confidence)
1. **Answer wrapper mismatch**
   - **Evidence:** Skill tool-interface output is an unwrapped
   ↪  string; contract requires `<answer>...</answer>`.
   - **Impact:** Even correct content may fail parsing / exact-match
   ↪  evaluation due to formatting.

2. **Potential non-compliance with "always emit an expression"**
   - **Evidence:** "otherwise unspecified" path in recovery rules.
   - **Impact:** Can produce invalid or missing output when
   ↪  localization is uncertain, violating the contract's completion
   ↪  requirement.

## 3. Preservation risks (static / rollout-free)
```

```
- **Exact-span preservation may be too strict:** Requiring exact
↪  punctuation/formatting increases the chance of mismatch if
↪  benchmark evaluation uses normalization (common in VQA-style
↪  benchmarks).
- **Dense table/grid ambiguity:** The artifact hazard list highlights
↪  ruled grids and tables; the skill procedure does not specify
↪  **explicit table-cell selection rules** beyond "smallest exact
↪  span" and "nearby similar strings" heuristic.
- **Handwritten/mixed layouts:** The procedure assumes reliable
↪  visual localization and "copy exactly"; with handwritten or mixed
↪  print, exact character-level extraction is a high-error mode.
- **Numeric/date/time extraction policy underspecified:** Operations
↪  mention date/time extraction, but no static rules cover common
↪  DocVQA variants (e.g., `12/05/2020` vs `May 12, 2020`, leading
↪  zeros, unit words).

## 4. Supported bounded edit recommendations (no skill rewrite, only
↪  bounded changes)
1. **Output wrapper alignment**
   - Update the skill's output specification to **always include
   ↪  `<answer>...</answer>`** as the outer container, matching the
   ↪  Task Contract IR.

2. **Remove/replace "otherwise unspecified"**
   - Modify recovery behavior so that it **always returns a non-empty
   ↪  answer expression** (even if low-confidence), staying within
   ↪  "one concise answer string/expression."

3. **Introduce minimal normalization guardrails (bounded)**
   - Add an explicit rule about permissible normalization consistent
   ↪  with DocVQA semantics already assumed in the IR recording
   ↪  (e.g., whitespace trimming; optionally case normalization),
   ↪  while still preferring grounded extraction.

4. **Add explicit table/grid disambiguation rule (bounded)**
   - For grid-like layouts: require selecting the value based on
   ↪  **row/column header alignment** (local evidence) rather than
   ↪  only "smallest exact span."

## 5. Findings that should NOT drive edits (evidence is weak)
- **"Prefer direct extraction over paraphrase"**: Generally aligned
↪  with DocVQA answer-from-document behavior; no clear contract
↪  conflict indicated.
- **"Compare nearby alternatives before finalizing"**: Reasonable and
↪  consistent with the task's visual-evidence grounding; no concrete
↪  mismatch signal from the IR.
- **List of operations (location/entity/date extraction, etc.)**:
↪  While potentially broad, it doesn't contradict the contract and
↪  may be harmless unless coupled with the output-format/non-empty
↪  constraints above.

---
```

Static synthesis: complete prompt

```
You are performing Rollout-Free Static Skill Synthesis for ROSE.

Benchmark: docvqa

Task: synthesize an updated frozen benchmark-level skill from the
↪  initial skill, code-style IR, and audit findings below.
```

```
Constraints:
- Use only the initial skill, IR record, and audit findings.
- Do not use held-out labels, hidden answers, correct labels,
↪  reference solutions, held-out verifier outcomes, held-out
↪  trajectories, or per-test rewriting.
- Preserve useful initial-skill procedures.
- Do not freely rewrite the skill.
- Apply only bounded, high-confidence edits supported by the audit.
- Prefer concise output/action contracts over broad checklists.
- Output only the final Markdown skill, with no surrounding
↪  explanation.

Initial skill:
```markdown
# DocVQA Skill

## Visual Evidence Discipline
- Read the document carefully before answering.
- Prefer the smallest exact text span that answers the question.
- When several nearby strings look similar, choose the one whose
↪  surrounding labels or layout best match the question.

## Exact Answer Discipline
- Copy names, numbers, and dates exactly from the document whenever
↪  possible.
- Prefer direct extraction over paraphrase.
- Before finalizing, compare the answer against nearby alternatives
↪  and keep the best-supported exact span.

```

Code-style IR record:
```markdown
## 1. Skill IR
- **tool_interface:** `input = {question, document-page image}`;
↪  `output = one concise answer string`
- **procedure:**
  1. Inspect the document page carefully.
  2. Use visible layout/text evidence to identify the answer target.
  3. Prefer the smallest exact text span that answers the question.
  4. For nearby similar strings, select the one best supported by
  ↪  surrounding labels/layout.
  5. Copy names, numbers, and dates exactly when possible.
- **preservation_rules:**
  - Keep answers grounded in the visible document image.
  - Prefer direct extraction over paraphrase.
  - Preserve exact spelling, punctuation, and formatting when
  ↪  extracting spans.
  - Compare nearby alternatives before finalizing.
- **recovery_rules:**
  - If multiple candidate spans are similar, fall back to the span
  ↪  with the strongest local layout/label support.
  - If the answer cannot be localized confidently, return the
  ↪  best-supported minimal exact span; otherwise unspecified.

## 2. Task Contract IR
- **intent:** answer the question from the attached DocVQA
↪  document-page image.
- **operations:** visible-evidence lookup; field binding; comparison;
↪  count; quantity extraction; date/time extraction; entity name
↪  extraction; location extraction; document-type identification.
- **output_contract:** emit exactly one concise answer inside
↪  `<answer>...</answer>`.
```
```

```
- **completion_requirement:** one image-grounded answer expression is
↪  emitted.

## 3. Artifact IR
- **artifact_type:** single document-page image
- **structure:** page-level document image paired with a question;
↪  observed layouts include ruled grids, horizontally ruled pages,
↪  multi-column pages, mixed layouts, sparse forms/notes, dense
↪  text/graphics, free text, tables/lists, handwritten forms, and
↪  diagrams.
- **locations:** question text is provided separately from the
↪  document image; evidence is located within the visible page
↪  image.
- **hazards:** layout ambiguity; visually similar nearby strings;
↪  dense tables/grids; mixed handwriting/printed content; no OCR or
↪  recognized text is guaranteed.
```

Audit findings:
```markdown
## 1. Detected static signals
- **Output formatting conflict risk:** Skill IR says `output = one
↪  concise answer string`, while Task Contract IR requires
↪  **wrapping the answer in `<answer>...</answer>`**.
- **Always-output constraint tension:** Skill "recovery_rules"
↪  includes: "**If the answer cannot be localized confidently...;
↪  otherwise unspecified.**" This can yield **empty/undefined
↪  output**, conflicting with "completion_requirement: one
↪  image-grounded answer expression is emitted."
- **Grounding vs. evaluation normalization uncertainty:** Skill
↪  preservation rules demand **exact
↪  spelling/punctuation/formatting**. DocVQA scoring often tolerates
↪  some formatting variance via normalization (exact details vary),
↪  so "exact span only" can become a **precision risk** if
↪  normalization is expected.
- **No OCR/recognition guaranteed (artifact hazard):** IR explicitly
↪  notes **no OCR/recognized text is guaranteed**, yet the procedure
↪  strongly relies on "smallest exact text span" extraction and
↪  "copy names/numbers/dates exactly."

## 2. Contract mismatch findings (high-confidence)
1. **Answer wrapper mismatch**
   - **Evidence:** Skill tool-interface output is an unwrapped
   ↪  string; contract requires `<answer>...</answer>`.
   - **Impact:** Even correct content may fail parsing / exact-match
   ↪  evaluation due to formatting.

2. **Potential non-compliance with "always emit an expression"**
   - **Evidence:** "otherwise unspecified" path in recovery rules.
   - **Impact:** Can produce invalid or missing output when
   ↪  localization is uncertain, violating the contract's completion
   ↪  requirement.

## 3. Preservation risks (static / rollout-free)
- **Exact-span preservation may be too strict:** Requiring exact
↪  punctuation/formatting increases the chance of mismatch if
↪  benchmark evaluation uses normalization (common in VQA-style
↪  benchmarks).
- **Dense table/grid ambiguity:** The artifact hazard list highlights
↪  ruled grids and tables; the skill procedure does not specify
↪  **explicit table-cell selection rules** beyond "smallest exact
↪  span" and "nearby similar strings" heuristic.
```

```
- **Handwritten/mixed layouts:** The procedure assumes reliable
↪  visual localization and "copy exactly"; with handwritten or mixed
↪  print, exact character-level extraction is a high-error mode.
- **Numeric/date/time extraction policy underspecified:** Operations
↪  mention date/time extraction, but no static rules cover common
↪  DocVQA variants (e.g., `12/05/2020` vs `May 12, 2020`, leading
↪  zeros, unit words).

## 4. Supported bounded edit recommendations (no skill rewrite, only
↪  bounded changes)
1. **Output wrapper alignment**
   - Update the skill's output specification to **always include
   ↪  `<answer>...</answer>`** as the outer container, matching the
   ↪  Task Contract IR.

2. **Remove/replace "otherwise unspecified"**
   - Modify recovery behavior so that it **always returns a non-empty
   ↪  answer expression** (even if low-confidence), staying within
   ↪  "one concise answer string/expression."

3. **Introduce minimal normalization guardrails (bounded)**
   - Add an explicit rule about permissible normalization consistent
   ↪  with DocVQA semantics already assumed in the IR recording
   ↪  (e.g., whitespace trimming; optionally case normalization),
   ↪  while still preferring grounded extraction.

4. **Add explicit table/grid disambiguation rule (bounded)**
   - For grid-like layouts: require selecting the value based on
   ↪  **row/column header alignment** (local evidence) rather than
   ↪  only "smallest exact span."

## 5. Findings that should NOT drive edits (evidence is weak)
- **"Prefer direct extraction over paraphrase"**: Generally aligned
↪  with DocVQA answer-from-document behavior; no clear contract
↪  conflict indicated.
- **"Compare nearby alternatives before finalizing"**: Reasonable and
↪  consistent with the task's visual-evidence grounding; no concrete
↪  mismatch signal from the IR.
- **List of operations (location/entity/date extraction, etc.)**:
↪  While potentially broad, it doesn't contradict the contract and
↪  may be harmless unless coupled with the output-format/non-empty
↪  constraints above.

---
```

Static synthesis: complete response

```
# DocVQA Skill

## Task Output Contract
- Always emit the final response as: `<answer>...</answer>`
↪  containing **exactly one** concise answer string.
- Never emit an empty/undefined output: if localization confidence is
↪  low, return the **best-supported** minimal answer span visible in
↪  the page.

## Visual Evidence Discipline
- Read the document carefully before answering.
- Use visible layout/text evidence in the provided document-page
↪  image to identify the answer target.
- Prefer the smallest exact text span that answers the question.
```

```
- When several nearby strings look similar, select the one whose
↪  surrounding labels or layout best match the question.

## Exact Answer Discipline (with minimal normalization)
- Copy names, numbers, and dates as they appear in the document
↪  whenever possible.
- Preserve exact spelling and punctuation when extracting spans.
- Apply only lightweight, semantics-preserving normalization to
↪  reduce formatting mismatches:
  - trim leading/trailing whitespace
  - collapse consecutive internal whitespace to a single space
- Before finalizing, compare the answer against nearby alternatives
↪  and keep the best-supported exact span (after the minimal
↪  normalization above).

## Table / Grid Disambiguation (bounded)
- If the page contains ruled grids/tables:
  - choose the cell whose row/column alignment best matches the
  ↪  question-relevant headers/labels (local evidence),
  - and then extract the smallest exact span within that chosen cell.

## Recovery / Low-Confidence Handling (no-empty rule)
- If multiple candidate spans are similar, fall back to the span with
↪  the strongest local layout/label support.
- If the answer cannot be localized confidently, still return the
↪  best-supported minimal span visible on the page (not
↪  "unspecified" or empty), following the same wrapper and minimal
↪  normalization rules.
```

Final frozen skill

```
# DocVQA Skill

## Task Output Contract
- Always emit the final response as: `<answer>...</answer>`
↪  containing **exactly one** concise answer string.
- Never emit an empty/undefined output: if localization confidence is
↪  low, return the **best-supported** minimal answer span visible in
↪  the page.

## Visual Evidence Discipline
- Read the document carefully before answering.
- Use visible layout/text evidence in the provided document-page
↪  image to identify the answer target.
- Prefer the smallest exact text span that answers the question.
- When several nearby strings look similar, select the one whose
↪  surrounding labels or layout best match the question.

## Exact Answer Discipline (with minimal normalization)
- Copy names, numbers, and dates as they appear in the document
↪  whenever possible.
- Preserve exact spelling and punctuation when extracting spans.
- Apply only lightweight, semantics-preserving normalization to
↪  reduce formatting mismatches:
  - trim leading/trailing whitespace
  - collapse consecutive internal whitespace to a single space
- Before finalizing, compare the answer against nearby alternatives
↪  and keep the best-supported exact span (after the minimal
↪  normalization above).

## Table / Grid Disambiguation (bounded)
- If the page contains ruled grids/tables:
```

```
  - choose the cell whose row/column alignment best matches the
  ↪  question-relevant headers/labels (local evidence),
  - and then extract the smallest exact span within that chosen cell.

## Recovery / Low-Confidence Handling (no-empty rule)
- If multiple candidate spans are similar, fall back to the span with
↪  the strongest local layout/label support.
- If the answer cannot be localized confidently, still return the
↪  best-supported minimal span visible on the page (not
↪  "unspecified" or empty), following the same wrapper and minimal
↪  normalization rules.
```