TerraWeave: GROUNDING POLICY-SKILL CO-EVOLUTION THROUGH VERIFIED EXECUTION FOR EARTH OBSERVATION AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Tool-augmented Earth observation (EO) agents solve multi-step geospatial tasks, yet successful tool execution does not guarantee that the dependencies required by downstream steps have been established. We introduce TerraWeave, which jointly grounds policy credit and skill evaluation in verified execution state. GeoState records observation-supported resources, relations, and readiness conditions. Grounded Dependency-Local Group Relative Policy Optimization (GDL-GRPO) assigns decision-local credit by normalizing verified progress among decisions sharing the same dependency opportunity. Skill candidates are compiled and execution-verified, then evaluated through frozen-policy ON/OFF continuations from the same executable snapshot to estimate incremental utility. Verified progress drives policy updates, while measured incremental utility selects skill versions for subsequent training. On the 1,169-task OpenEarthAgent test set, TerraWeave reaches 50.37/77.01 in native answer score (Ans.) and generation-tool execution (Gen.). Under matched policy-training rollout allowances, it improves Ans./Gen. by 2.26/2.76 percentage points over skill-augmented RL baseline ReSkill and by 7.87/12.18 points over Standard GRPO. With the bank disabled, the co-evolved policy retains most gains over policy-only GDL-GRPO; enabling the bank adds runtime gains.

1 INTRODUCTION

Earth observation (EO) agents solve multi-step geospatial tasks by chaining tools for imagery interpretation, spectral analysis, raster processing, and GIS operations (Shabbir et al., 2026; Feng et al., 2026c; Shabbir et al., 2025; Chen et al., 2024). These stateful workflows depend on persistent resources with correct spatial, temporal, and semantic bindings: a layer must cover the requested region, an index must retain its acquisition date, and an output map must contain every requested category. Consequently, a successful tool call need not establish downstream dependencies. An area extent, for example, does not provide boundary data, and a hospital–school task remains incomplete until both point layers, a valid distance, and the requested overlay are established.

The learning problem is therefore twofold: assign credit to decisions that advance unresolved dependencies, and determine which procedural guidance is useful under an evolving policy. RL methods used in agent training assign credit at the level of trajectories, states, turns, or shared transitions (Shao et al., 2024; Feng et al., 2026b; Li et al., 2026; Tao et al., 2026). In stateful EO workflows, however, the comparison unit matters: the same tool can face different prerequisites, while different actions can establish the same requirement. Skill systems likewise retrieve and revise procedural guidance as policies learn (Xia et al., 2026; He et al., 2026; Tu et al., 2026; Wang et al., 2026), but policy updates also change the execution states in which that guidance is applied. Thus, policy credit and skill evaluation depend on each decision's execution conditions.

TerraWeave uses verified execution state as a shared reference for policy credit and skill evaluation (Fig. 1). GeoState records observation-supported resources, relations, and readiness conditions as tool interactions unfold; it represents what execution has established rather than prescribing a fixed solution trajectory. For policy learning, Grounded Dependency-Local Group Relative Policy Optimization (GDL-GRPO) compares decisions at matched dependency opportunities—compatible

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

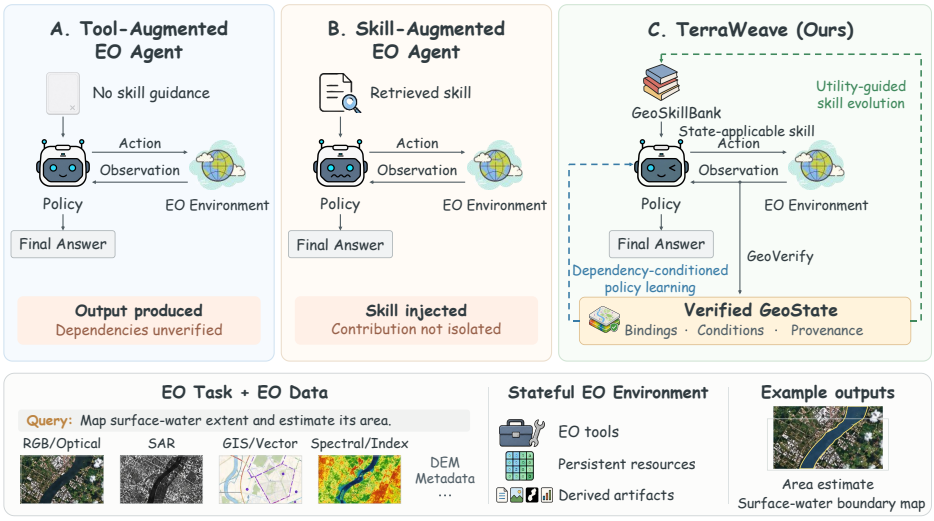

Figure 1: **Grounding policy-skill co-evolution in verified execution.** (A) Tool execution can produce outputs without establishing downstream dependencies. (B) Retrieved skills can accompany successful execution without a saved-state test of their incremental effect. (C) TerraWeave uses GeoState as a shared verified reference for dependency-conditioned policy learning and utility-guided skill evolution. Bottom: representative EO inputs, persistent resources, and outputs.

unmet requirements and execution readiness—so equivalent verified progress can be compared even when it occurs at different trajectory positions or through different actions. For skill evaluation, paired continuations restore the same executable snapshot under a frozen policy and vary only whether a candidate version is delivered, estimating its incremental utility. The same verified executions also support skill proposals and revisions, while measured utility determines which versions guide subsequent policy training.

On the 1,169-task OEA-TEST benchmark, TerraWeave reaches 50.37/77.01 in native answer score and generation-tool execution, improving over Standard GRPO by 7.87/12.18 percentage points under the same policy-training rollout allowance. Crossed evaluation further separates learned policy gains from runtime assistance: most of the improvement beyond policy-only GDL is retained with the bank disabled, while runtime skill delivery provides an additional 1.18/1.15-point gain.

Our contributions are:

- **A shared verified EO execution reference.** GeoState records observation-supported resources, relations, and readiness. GeoState jointly grounds dependency comparison and snapshot-based skill evaluation through the same verification semantics.
- **Dependency-conditioned policy credit.** GDL-GRPO assigns local credit among decisions with aligned unmet requirements and execution conditions, so equivalent progress can be compared across trajectory positions or producing tools.
- **Execution-validated policy-skill co-evolution.** Verified executions supply policy feedback and the records from which skill candidates are proposed. Compilation checks executability. Verified admission checks whether the declared consequence occurs. Frozen-policy paired continuations estimate incremental utility. Measured utility, rather than the admission record, selects versions for later training.

2 Related Work

Earth observation and geospatial agents. Tool-augmented language agents interleave reasoning with external actions, from prompted reason–act interaction to learned or retrieved API use while EO agents extend this paradigm to multi-step visual, spectral, raster, and GIS tool execution (Yao

et al., 2022; Schick et al., 2023; Qin et al., 2024; Chen et al., 2024; Shabbir et al., 2025; 2026; Feng et al., 2026c). Precondition–effect representations make tool dependencies explicit (Fikes & Nilsson, 1971; McDermott et al., 1998; Xu et al., 2026). GeoAgentBench and UnivEARTH specialize broader interactive-agent evaluation in executable geospatial reasoning (Liu et al., 2024; Zhou et al., 2024; Yao et al., 2024; Yu et al., 2026; Kao et al., 2026). RSMeM combines hierarchical domain knowledge with failure-refined execution experience for multi-step tool use (Wu et al., 2026). TerraWeave asks how observation-verified execution state can support both policy-credit comparison and skill-intervention evaluation in these workflows.

Credit assignment for tool-using agents. PPO and GRPO optimize policies from return-based advantages (Schulman et al., 2017; Shao et al., 2024); WebRL and ReTool adapt outcome-driven RL to interactive tool use (Qi et al., 2025; Feng et al., 2026a). Recent methods localize credit at transitions, turns, states, or trajectory-graph edges (Luo et al., 2025; Tao et al., 2026; Feng et al., 2026b; Li et al., 2026). GiGPO and SALT organize comparisons through encountered states or shared transitions. GDL-GRPO instead matches decisions by unmet requirements and execution readiness, aligning verified progress across positions or actions that establish the same dependency.

Procedural memory and skill evolution. Procedural memory reuses successful or reflective experience (Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2024; Ouyang et al., 2026), while executable skill systems store or compose behaviors grounded in environment feedback (Wang et al., 2023; Ahn et al., 2022; Klissarov et al., 2025). Skill evolution, libraries, and internalization couple reusable procedures with policy learning (Xia et al., 2026; Wang et al., 2026; Lu et al., 2026; Shi et al., 2026). Closely related systems evaluate skill usefulness through shared-policy version comparisons, guided/unguided contrasts, fixed-context marginal utility, or execution-based verification and refinement (He et al., 2026; Tu et al., 2026; Zhang et al., 2026; Yang et al., 2026). TerraWeave separates realization from incremental utility: compilation and admission check executability and declared consequences, while frozen-policy ON/OFF continuations estimate delivery effects from the same executable snapshot. GeoState also supplies the dependency opportunities for policy credit, coupling both evaluations under shared execution semantics.

3 TerraWeave

3.1 Problem Setup

We consider an EO task q with public initial resources R_0, a stateful tool environment $\mathcal{E}$, a policy π_θ, and a procedural skill bank B. At step t, the policy chooses $a_t \sim \pi_\theta(\cdot \mid q, R_0, h_t, c_t)$ from visible history h_t and optional guidance c_t, then receives $o_{t+1} \leftarrow \mathcal{E}(a_t)$.

The requirement graph G_q has nodes for task-required resources, relations, or postconditions, while directed edges encode prerequisite dependencies. It is deterministically instantiated from the public task and initial resources using task and tool contracts frozen before training. GeoVerify checks executed actions and returned observations against G_q to maintain GeoState. GeoState remains verifier-side: the policy acts only on its visible interaction context and any guidance supplied to it.

TerraWeave uses this verified execution state to organize policy credit and skill interventions, coupling policy learning with bank evolution (Fig. 2).

3.2 Verified State and Comparison Views

One verified execution point supports two views: a dependency comparison key κ_t for policy credit and an executable intervention snapshot ζ_t for skill evaluation.

GeoState. GeoVerify maintains the observation-grounded execution state

$$s_t = (R_t, z_t, P_t, \xi_t), \tag{1}$$

where R_t contains typed resource records and bindings, z_t contains verified semantic and spatiotemporal relations, P_t contains readiness, artifact, and endpoint conditions, and ξ_t records the observations supporting each established fact.

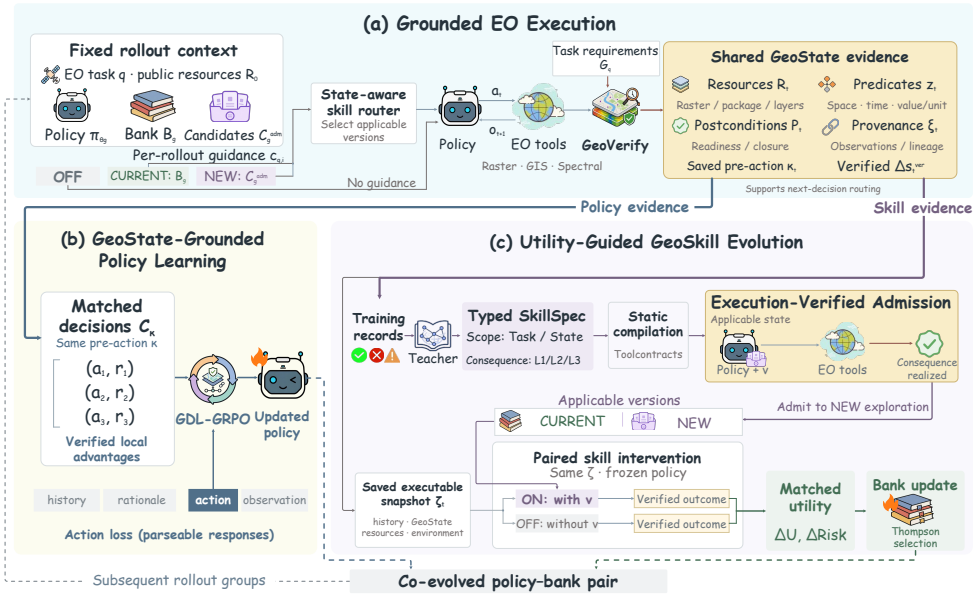

Figure 2: **Overview of TerraWeave.** (a) GeoVerify converts tool observations into verified GeoState evidence. (b) Policy learning assigns local credit by comparing decisions made under the same dependency opportunity. (c) GeoSkill candidates pass compilation and verified admission before paired continuations measure their incremental utility. Policy updates and skill-bank updates then form the co-evolution loop.

A resource or condition is established only with environment evidence. After action a_t returns o_{t+1}, the update $\Delta s_t^{\text{ver}} = \text{Verify}_q(s_t, a_t, o_{t+1})$ records satisfaction of additional requirements, invalidates stale evidence, or leaves the logical state unchanged, with provenance updated alongside it.

Two views of one verified execution point. Let $\text{Sat}_q(s_t)$ denote the requirements in G_q currently satisfied by verified evidence. The frontier F_t contains the unmet requirements whose parents already belong to $\text{Sat}_q(s_t)$. A fixed topological and semantic-role order selects an active target $r_t^\star$. Frontier-wide readiness $\rho_t^F = \{(r, \rho_t(r)) : r \in F_t\}$ records canonical resource roles and execution preconditions for every frontier requirement: the active target does not restrict an action from advancing another requirement in F_t.

The resulting views are:

$$\begin{aligned} \kappa_t &= (q, F_t, r_t^\star, \rho_t^F), && \text{dependency comparison key,} \\ \zeta_t &= (h_t^{\text{vis}}, s_t, \mathcal{X}_t, \mathcal{M}_t), && \text{executable intervention snapshot.} \end{aligned} \tag{2}$$

Here h_t^{vis} is the visible interaction history, $\mathcal{X}_t$ contains the resource contents and versions needed to resume execution, and $\mathcal{M}_t$ contains the restorable tool/runtime state.

We save κ_t before action generation; subsequent outcomes never relabel this comparison key. Decisions share it only when the frontier, the active target, and frontier-wide readiness agree. Readiness uses semantic identity, including source version and lineage. Appendix B details requirement-graph construction, verification, semantic comparison matching, and executable snapshot restoration.

3.3 Dependency-Conditioned Policy Credit

GDL-GRPO turns verified execution progress into decision-local advantages within matched dependency opportunities. Each rollout group g is generated under a fixed behavior policy π_{θ_g}.

Verified local reward. The reward measures verified task progress rather than tool success alone. Let $\omega_{i,t}$ indicate that decision (i,t) yields a usable execution record. The opportunity reward is

$$r_{i,t}^{\text{opp}} = \omega_{i,t}\left(c_{i,t}^{\text{loc}} + c_{i,t}^{\text{VN}} - p_{i,t}\right) + c_{i,t}^{\text{end}}. \tag{3}$$

Here $c_{i,t}^{\text{loc}} = \min\{C_{\text{dec}}, c_{i,t}^{\text{raw}}\}$ caps process credit, c^{raw} records net-new verified task progress and readiness transitions, c^{VN} rewards independent evidence that changes the valid plan, c^{end} rewards grounded task completion, and p penalizes controllable failures.

Repeated support for an already established condition receives no new progress credit, and environment failures outside the policy's control do not define a policy-learning opportunity.

Matched local advantage. Within rollout group g, let $\mathcal{C}_\kappa^{(g)}$ contain eligible decisions with the same saved κ, taking the earliest eligible decision from each trajectory. For the opportunity-reward mean $\mu_\kappa^{(g)}$ and standard deviation $\sigma_\kappa^{(g)}$,

$$A_{i,t}^{\text{GDL}} = \frac{r_{i,t}^{\text{opp}} - \mu_{\kappa_{i,t}}^{(g)}}{\sigma_{\kappa_{i,t}}^{(g)} + \epsilon}. \tag{4}$$

Normalization requires a non-degenerate set, distinct-rollout support, and a verified positive-progress anchor. For controllable failures we use $\widetilde{A}_{i,t} = \min(0, A_{i,t}^{\text{GDL}})$; otherwise $\widetilde{A}_{i,t} = A_{i,t}^{\text{GDL}}$. GDL-GRPO provides local comparative credit, not long-horizon state-value estimation: a shared κ_t marks a comparable opportunity, not a predictive state or equal future return.

Action-token optimization. For parseable responses, GDL-GRPO applies $\widetilde{A}_{i,t}$ to current action tokens using a clipped PPO objective with fixed-reference KL; history, observations, and delivered guidance remain conditioning context. Appendix C gives the full objective, event semantics, malformed-response support, and update/eligibility rules.

3.4 GeoSkill Admission and Utility

A GeoSkill version specifies when guidance applies, which actions it recommends, and an observable consequence that tool observations can verify. Each immutable version $v = (\gamma, \ell, \chi; \texttt{when}, \texttt{act}, \texttt{verify}, \texttt{repair}, \texttt{stop})$ records delivery scope γ, consequence level ℓ, and supporting evidence χ. Consequences cover resource/action transitions (L1), semantic, numerical, or unit correctness (L2), and endpoint/artifact completion (L3). Task skills apply from initialization; State skills apply at eligible dependency frontiers.

A teacher proposes candidates from verified training records. *Compilation* checks tools, argument roles, and resource dependencies against the public contracts. *Execution-verified admission* requires one independent continuation to realize the declared consequence. Admission establishes consequence realization; *matched utility evaluation* estimates the incremental effect of delivering the candidate. Admission evidence is not reused as utility evidence, and utility probes do not update the policy. Routing restricts delivery to applicable, safe versions; CURRENT selection additionally requires paired utility evidence.

Matched utility. At utility window w, we freeze an anchor policy $\bar{\theta}_w$, candidate versions, and a set of eligible executable intervention snapshots. For each version v, ON and OFF branches restore separate copies of the same snapshot ζ_t, use the same continuation seed and verifier, and differ only in whether v is delivered. Writable outputs are kept separate across the two branches.

Incremental utility is the mean ON–OFF difference in realization of the declared consequence. For selection, utility is adjusted for the ON–OFF difference in execution risk after branching. Both estimates are specific to the frozen anchor and enrolled snapshots. Appendix D gives the utility estimators, evaluation protocol, selection rules, and full GeoSkill lifecycle.

3.5 Two-Timescale Co-Evolution

Because skill utility depends on the policy, TerraWeave updates the policy frequently but evaluates skills periodically under frozen anchors. Verified executions supply policy feedback and evidence for

a teacher to propose or revise candidates. Within each interval, GDL-GRPO uses skill-free (OFF), supported-bank (CURRENT), and safely admitted candidate (NEW) rollouts. At each scheduled boundary, the current policy becomes the frozen utility anchor; paired continuations from executable snapshots evaluate admitted versions before the bank update.

At each boundary, a task-clustered Bayesian bootstrap estimates anchor-specific utility distributions. Thompson sampling selects applicable CURRENT versions with sufficient paired support; under-evaluated candidates remain exploratory (NEW). The bank and its anchor-specific utility distributions stay fixed during the next training interval.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

OEA-TEST reports graded Ans. on 1,024 non-generation tasks and generation-tool success (Gen.) on 145 generation tasks. HELD-OUT uses Binary Answer, mean verified non-endpoint completion (VSC), grounded endpoint completion (GEC), and FRR on 591 common blocked states.

Controlled comparisons match the cold start and 4,328 policy-training trajectories. TerraWeave also uses no-gradient admission and utility probes, leaving total execution cost unmatched (Appendices A and E.11).

Independent, method- and arm-blind GeoVerify audits yield 98.0% endpoint agreement (147/150; Appendix B.5).

4.2 TASK PERFORMANCE AND ROBUSTNESS

On OEA-TEST, TerraWeave gains 7.87/12.18 percentage points (pp) over Standard GRPO and 2.26/2.76 pp over ReSkill in Ans./Gen., reaching 50.37 Ans. and 77.01 Gen. under matched policy-training rollout allowances. Table 1 provides broader capability context: all three tool-order scores improve over Standard GRPO, while tool-selection F1 is mixed against the EO-specialized model.

Table 1: **Capability profiles on OEA-TEST.** Param. gives total model sizes; "–" denotes unavailable values. Local results average three decoding seeds, using prespecified primary runs for trained systems. Scores (%) use metric-specific eligibility; bold/underlining mark best/second-best. Score sources, parameter sources, and aggregation are specified in Appendix E.1. Per./Op. denote Perception/Operation; AnyOr., SameO., and Uni. compare order-agnostic tool multisets, exact sequences, and unique-tool sets, respectively.

Model	Param.	Tool-selection F1				Tool order			Native scores	
		Per.	Op.	Logic	GIS	AnyOr.	SameO.	Uni.	Ans.	Gen.
Frontier LLMs										
DeepSeek-V4-Pro	1.6T	39.71	54.45	33.70	92.70	59.71	59.11	62.87	47.80	74.48
GPT-5.5	–	30.02	57.64	8.06	96.11	47.22	47.22	49.10	<u>47.97</u>	66.90
GLM-5.2	744B	43.01	<u>58.12</u>	37.31	93.13	59.79	59.20	64.84	46.43	<u>76.55</u>
Kimi-K2.6	1T	31.02	27.40	33.58	92.50	57.14	56.72	60.39	40.32	68.28
Open-source LLMs										
Qwen3-4B-2507	4B	13.39	29.03	6.93	59.72	6.67	5.56	8.56	6.66	11.72
Qwen2.5-7B	7B	19.13	37.78	29.23	76.93	32.42	31.39	37.64	14.83	46.21
Llama-3.1-Instruct	8B	22.54	16.45	32.03	64.64	39.01	37.72	44.91	12.70	55.17
Qwen3-32B	32B	28.50	31.46	10.05	47.53	4.02	4.02	5.47	7.12	22.76
Qwen3.5-9B	9B	25.48	26.20	14.80	74.48	16.11	14.11	16.34	15.82	39.08
Published EO-specialized reference										
OpenEarthAgent-4B	4B	**58.30**	56.76	**51.18**	**98.52**	<u>67.75</u>	<u>67.24</u>	<u>72.71</u>	45.26	75.86
Qwen3.5-9B training comparison										
Standard GRPO	9B	33.47	49.83	31.96	86.84	47.16	46.03	51.48	42.50	64.83
TerraWeave (Qwen3.5-9B)	9B	<u>53.05</u>	**60.68**	<u>45.16</u>	<u>96.98</u>	**68.72**	**67.78**	**72.97**	**50.37**	**77.01**

The gains cover answer quality and successful artifact production. Table 2 provides the controlled training comparisons; paired 95% intervals for the ReSkill differences exclude zero (Appendix Table 18).

Table 2: **Credit assignment and skill co-evolution under matched policy-training rollout allowances.** Panel (a) shares verification feedback and training settings; (b) additionally matches teacher request caps. The table compares matched policy-training exposure, not total-compute efficiency; extra no-update probes are counted separately. Primary-run OEA-TEST scores (%) average three decoding seeds. Closest-method paired intervals appear in Appendix Table 18.

Method	Ans.↑	Gen.↑
(a) Same-feedback credit assignment		
GRPO + GeoVerify	44.20	67.13
GiGPO + GeoVerify	44.62	67.82
SALT + GeoVerify	44.91	68.28
GDL-GRPO θ_P (Bank OFF)	**46.04**	**69.43**
(b) EO-adapted skill-RL systems		
D2Skill	47.84	74.02
ReSkill	48.11	74.25
TerraWeave	**50.37**	**77.01**

Across three additional training runs, TerraWeave improves over ReSkill by 2.76 Ans. and 3.68 Gen. points on average. On a 150-task compositional holdout from RESERVE, TerraWeave raises the number of correct answers from ReSkill's 63 to 74 (+7.3 pp). Appendix E.11 details both evaluations. The repeated runs support the native gains; the compositional holdout extends them to unseen combinations of familiar tools and requirements.

4.3 Dependency-Conditioned Policy Credit

We test whether dependency-aware credit helps decisions advance unmet EO requirements. With skills off and GeoVerify feedback fixed, it raises Ans./Gen. by 1.84/2.30 pp over trajectory-level aggregation (Table 2a). Against SALT, the +1.13/+1.15 pp native-score gaps are non-decisive: paired 95% intervals include zero (Appendix Table 18). The VPR and (q, t) analyses below test dependency-conditioned credit more directly.

Table 3: **Constructive decomposition and mechanism ablations.** (a) Adds grounded credit, co-evolution, then bank delivery; (b) evaluates targeted operational replacements. Binary Answer, VSC, and GEC use 297 HELD-OUT tasks; FRR uses 591 common blocked states with whole-bank delivery OFF/ON.

(a) Constructive decomposition

Configuration	Grounded credit	Co-evolution	Runtime bank	Binary Answer	VSC (297)	GEC (297)	FRR (591)
Standard GRPO	–	–	–	41.4	48.7	9.4	10.0
GDL-GRPO θ_P	✓	–	OFF	45.5	56.8	14.1	16.1
Co-evolved θ_C	✓	✓	OFF	48.8	63.6	18.5	21.5
TerraWeave	✓	✓	ON	**50.5**	**69.7**	**24.6**	**29.4**

(b) Targeted mechanism ablations

Variant	Replacement	Binary Answer	VSC (297)	GEC (297)	FRR (591)
Full TerraWeave	–	**50.5**	**69.7**	**24.6**	**29.4**
w/o GeoState	History + trajectory GRPO	46.1	50.9	12.5	13.5
w/o dependency-opportunity matching	(q, t) grouping	47.1	59.7	16.5	19.0
First-version Bank	First admitted version	48.5	64.3	19.9	23.0
w/o matched utility	Uniform identity; newest version	48.1	62.3	18.5	20.5
w/o compiler / verified admission	Sanitize, then direct NEW	48.1	62.8	18.2	18.4

Within eligible common snapshots, GDL's verified progress rate (VPR) advantage over trajectory-level credit grows from 2.6 to 6.5 to 11.6 pp across remaining depths 1–2, 3–4, and ≥ 5, while

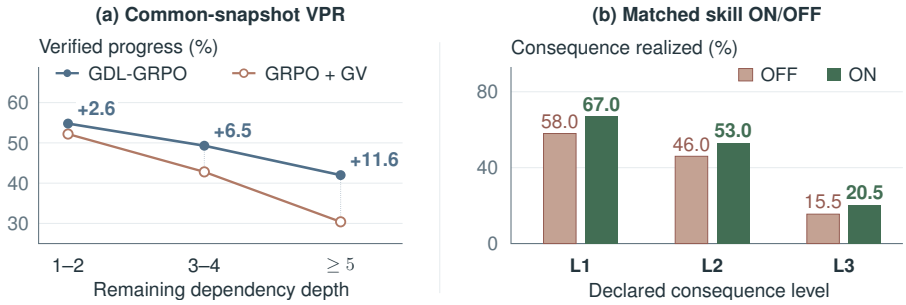

Figure 3: **Dependency-conditioned credit and skill delivery improve execution.** (a) Common-snapshot VPR (12,667 snapshots; skills OFF; GV: GeoVerify). Eligible/collected coverage is 86/78/64% by depth. (b) Paired rates on 100/150/200 UIDs, four matched pairs each (L1 resource/action, L2 semantic/unit, L3 endpoint/artifact). Task pairs repeat one initialization under four seeds; State pairs use distinct frontiers (Appendix Table 29).

eligibility coverage falls from 86% to 64% (Fig. 3a). This pattern is consistent with dependency-local credit helping longer unresolved workflows. Replacing κ with task-and-step grouping (q, t) in the full co-evolution system reduces GEC from 24.6% to 16.5% while retaining local rewards and loss eligibility (Table 3b). Together, these results support grouping decisions by verified dependency conditions rather than by trajectory position alone.

4.4 Policy-Skill Co-Evolution

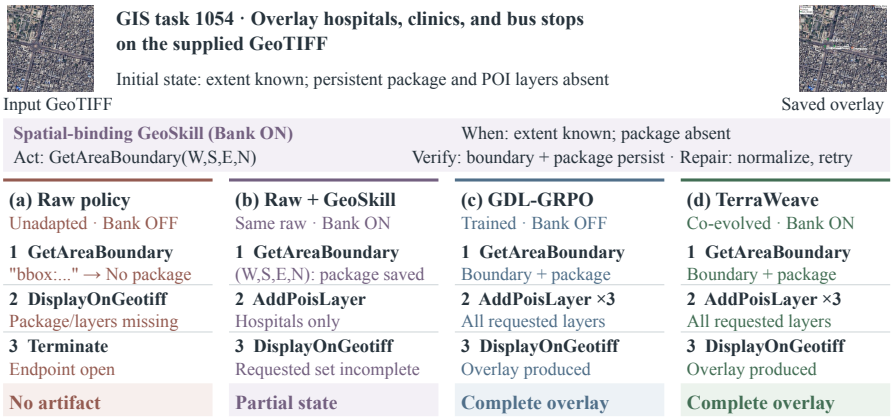

Figure 4: **Prerequisite repair is not end-to-end completion.** On GIS task 1054, the raw policy fails to create the spatial package; adding a spatial-binding GeoSkill repairs the package but leaves the requested POI set incomplete. GDL-GRPO without runtime guidance and full TerraWeave both deliver the requested overlay. Numbers index action groups; matched skill-delivery effects appear in Fig. 3b and Appendix E.7.

Table 3a decomposes grounded execution on HELD-OUT. GEC rises $9.4 \rightarrow 14.1 \rightarrow 18.5 \rightarrow 24.6\%$ across Standard GRPO, GDL-GRPO, the co-evolved policy with the bank disabled, and full TerraWeave. Across these configurations, correct answers with all pre-answer dependencies verified rise $71 \rightarrow 98 \rightarrow 120 \rightarrow 134$ of 297 tasks; those without closure fall from 52 to 16. Answer gains increasingly coincide with verified dependency closure.

In crossed OEA-TEST evaluation (Table 4a), co-evolved θ_C with the bank OFF exceeds policy-only θ_P by 3.15/6.44 pp in Ans./Gen. Restoring the bank adds 1.18/1.15 pp. Most native gains beyond policy-only GDL-GRPO are retained in the policy, with additional runtime assistance. The crossed evaluation separates retained policy gains from runtime assistance; it does not attribute the retained gain to any single bank-lifecycle operation.

The bundled w/o GeoState replacement reduces GEC most, from 24.6% to 12.5% (Table 3b). Removing compilation and execution-verified admission lowers FRR from 29.4% to 18.4%, supporting executable candidate screening for recovery. Keeping the first admitted version lowers GEC to 19.9%, supporting version revision; removing matched utility lowers it to 18.5%, supporting utility-guided bank management.

Table 4: **Runtime bank gains and cross-policy transfer.** Both panels use the same frozen bank $B^\star$: (a) crosses it with the learned policies; (b) attaches it to other fixed policies. Comparing OFF rows shows retained policy gains; $\Delta_B = \mathrm{ON} - \mathrm{OFF}$ measures runtime bank assistance. Differences (pp) use native aggregation (Appendix E.1). Paired Ans. intervals for panel (a) appear in Appendix Table 26.

(a) Crossed policy-bank evaluation

Policy	Ans. OFF	Ans. $+B^\star$	Δ_B	Gen. OFF	Gen. $+B^\star$	Δ_B
θ_P	46.04	46.84	+0.80	69.43	70.34	+0.92
θ_C	49.19	50.37	**+1.18**	75.86	77.01	**+1.15**

(b) Frozen target transfer, Δ_B (pp)

Target policy	ΔAns.	ΔGen.	Target policy	ΔAns.	ΔGen.
DeepSeek-V4-Pro	+4.03	+1.38	Qwen3-32B	+0.74	−0.46
GPT-5.5	+2.89	+2.53	OpenEarthAgent-4B	+0.92	+1.15
GLM-5.2	+0.35	+0.69	Qwen3.5-9B	−0.15	−0.23
Kimi-K2.6	+0.30	+0.46	Qwen3-4B raw	−0.69	0.00
Qwen2.5-7B	+0.38	+0.92			

Matched effects of skill delivery. Matched interventions evaluate the incremental effect of skill delivery. Versions are selected before HELD-OUT outcomes are observed. From the same executable intervention snapshot under a frozen policy, ON delivery raises L1/L2/L3 consequence realization by 9.0/7.0/5.0 pp; all three paired 95% intervals exclude zero (Fig. 3b).

Recovery. In a separate recovery-specific panel, State guidance is tested on 591 matched pairs from 207 tasks, holding the policy and Task guidance fixed. It raises frontier restoration from 18.3% to 23.4% and repair-to-endpoint completion from 13.9% to 18.1% (+4.2 pp). The gains extend from repairing blocked dependencies to completing the requested workflow.

Transfer. Frozen-bank transfer tests reuse by policies outside co-evolution without parameter updates. Reuse is receiver-dependent: some policies improve, while others see limited or negative gains (Table 4b). Appendix E.4–E.8 details dependency closure, paired effects, transfer, and recovery.

5 Conclusion and Limitations

TerraWeave uses observation-verified execution state to couple dependency-conditioned policy credit with matched skill-utility evaluation. Under shared GeoVerify feedback, dependency-local credit improves grounded execution relative to trajectory-level aggregation. The co-evolved policy retains most gains beyond policy-only GDL-GRPO with the bank disabled, while runtime skill delivery adds further gains. These results suggest that verified execution state can both check execution success and guide which decisions and procedural interventions influence learning.

The current study has several boundaries. Tool-contract verification establishes execution consistency, but does not independently validate the scientific accuracy of EO products, and the frozen task templates require maintenance as tools and contracts evolve. Matched skill interventions additionally require restorable resources and runtime state, and incur extra no-gradient executions. Skill reuse is also policy-dependent: the frozen bank benefits some target policies but yields limited or negative gains for others. Finally, grounded completion remains zero on the 11 complex/rare HELD-OUT tasks, leaving these evaluated workflows unresolved. Extending these abstractions beyond EO will therefore require explicit task contracts and domain-appropriate verification semantics.

AI USE STATEMENT

We used generative AI tools for manuscript drafting and editing, literature assistance, and feedback on experimental methodology and result interpretation. Using author-provided content, the tools helped improve clarity, organization, terminology, and consistency across the paper and appendices. Literature assistance covered search queries, relevant studies, and reference organization. Methodological feedback covered control experiments, evaluation protocols, quantities, sets, reference distributions, evidence utility, invalid-evidence responses, ownership, statistical interpretation, and the scope of conclusions.

The authors reviewed and revised all retained AI-assisted material, checked methodological descriptions and reported values, and verified references and citation relevance against primary sources. They made the final analytical decisions and take responsibility for the manuscript, references, experimental results, and scientific claims.

REPRODUCIBILITY STATEMENT

Appendices A–E document data, training, optimization, verification, the skill lifecycle, evaluation, intervention enrollment, statistical units, independent training runs, and execution accounting. We will publicly release code and trained model weights.

REFERENCES

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.

Yuxing Chen, Weijie Wang, Sylvain Lobry, and Camille Kurtz. An llm agent for automatic geospatial data analysis. *arXiv preprint arXiv:2410.18792*, 2024.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms. In *International Conference on Learning Representations*, volume 2026, pp. 37909–37926, 2026a.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *Advances in Neural Information Processing Systems*, 38:46375–46408, 2026b.

Peilin Feng, Zhutao Lv, Junyan Ye, Xiaolei Wang, Xinjie Huo, Jinhua Yu, Wanghan Xu, Wenlong Zhang, Lei Bai, Conghui He, et al. Earth-agent: Unlocking the full landscape of earth observation with agents. In *International Conference on Learning Representations*, volume 2026, pp. 55700–55764, 2026c.

Richard E Fikes and Nils J Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208, 1971.

Zelin He, Haotian Lin, Boran Han, Wei Zhu, Haoyang Fang, Bernie Wang, Xuan Zhu, Runze Li, and Matthew Reimherr. Reskill: Reconciling skill creation with policy optimization in agentic rl. *arXiv preprint arXiv:2606.01619*, 2026.

Chia Hsiang Kao, Wenting Zhao, Cheryl Lam, Aarush Umap, Shreelekha Revankar, Samuel Speas, Snehal Bhagat, Rajeev Datta, Cheng Perng Phoo, Utkarsh Mall, et al. Towards llm agents for earth observation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 2597–2611, 2026.

Martin Klissarov, Mikael Henaff, Roberta Raileanu, Shagun Sodhani, Pascal Vincent, Amy Zhang, Pierre-Luc Bacon, Doina Precup, Marlos C Machado, and Pierluca D'Oro. Maestromotif: Skill design from artificial intelligence feedback. In *International Conference on Learning Representations*, volume 2025, pp. 18029–18057, 2025.

Jiazheng Li, Yawei Wang, Qiaojing Yan, Yijun Tian, Zhichao Xu, Huan Song, Panpan Xu, and Lin Lee Cheong. Salt: Step-level advantage assignment for long-horizon agents via trajectory graph. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 4709–4725, 2026.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, volume 2024, pp. 52989–53046, 2024.

Zhengxi Lu, Zhiyuan Yao, Jinyang Wu, Chengcheng Han, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Skill0: In-context agentic reinforcement learning for skill internalization. *arXiv preprint arXiv:2604.02268*, 2026.

Xufang Luo, Yuge Zhang, Zhiyuan He, Zilong Wang, Siyun Zhao, Dongsheng Li, Luna K Qiu, and Yuqing Yang. Agent lightning: Train any ai agents with reinforcement learning. *arXiv preprint arXiv:2508.03680*, 2025.

Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. PDDL—The Planning Domain Definition Language, Version 1.2. Technical Report CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control, 1998.

Siru Ouyang, Jun Yan, I Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long Le, Samira Daruki, Xiangru Tang, et al. Reasoningbank: Scaling agent self-evolving with reasoning memory. In *International Conference on Learning Representations*, volume 2026, pp. 94327–94354, 2026.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Jiadai Sun, Xinyue Yang, Yu Yang, Shuntian Yao, Wei Xu, Jie Tang, and Yuxiao Dong. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 79791–79821, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/c66e1fcc9691aae706250638f36f681b-Paper-Conference.pdf.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *International Conference on Learning Representations*, volume 2024, pp. 9695–9717, 2024.

Donald B Rubin. The bayesian bootstrap. *The annals of statistics*, pp. 130–134, 1981.

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.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Akashah Shabbir, Muhammad Akhtar Munir, Akshay Dudhane, Muhammad Umer Sheikh, Muhammad Haris Khan, Paolo Fraccaro, Juan Bernabe Moreno, Fahad Shahbaz Khan, and Salman Khan. Thinkgeo: Evaluating tool-augmented agents for remote sensing tasks. *arXiv preprint arXiv:2505.23752*, 2025.

Akashah Shabbir, Muhammad Umer Sheikh, Muhammad Akhtar Munir, Hiyam Debary, Mustansar Fiaz, Muhammad Zaigham Zaheer, Paolo Fraccaro, Fahad Shahbaz Khan, Muhammad Haris Khan, Xiao Xiang Zhu, et al. Openearthagent: A unified framework for tool-augmented geospatial agents. In *European Conference on Computer Vision*, pp. 172–191. Springer, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Yaorui Shi, Yuxin Chen, Zhengxi Lu, Yuchun Miao, Shugui Liu, Qi Gu, Xunliang Cai, Xiang Wang, and An Zhang. Skill1: Unified evolution of skill-augmented agents via reinforcement learning. *arXiv preprint arXiv:2605.06130*, 2026.

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.

Leitian Tao, Baolin Peng, Wenlin Yao, Tao Ge, Hao Cheng, Mike Hang Wang, Jianfeng Gao, and Sharon Li. Trace: Turn-level reward assignment via credit estimation for long-horizon agents. *arXiv preprint arXiv:2607.13988*, 2026.

Songjun Tu, Chengdong Xu, Qichao Zhang, Yaocheng Zhang, Xiangyuan Lan, Linjing Li, Dong Li, and Dongbin Zhao. Dynamic dual-granularity skill bank for agentic rl. *arXiv preprint arXiv:2603.28716*, 2026.

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

Jiongxiao Wang, Qiaojing Yan, Yawei Wang, Yijun Tian, Soumya Smruti Mishra, Zhichao Xu, Megha Gandhi, Panpan Xu, and Lin Lee Cheong. Reinforcement learning for self-improving agent with skill library. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1529–1550, 2026.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. *arXiv preprint arXiv:2409.07429*, 2024.

Bingxian Wu, Yu Zhang, Zonghao Guo, Tang Liu, Chen Qian, Yuxiang Lu, Xingbo Du, Yanghao Li, Yidan Zhang, Chi Chen, et al. Rsmem: Knowledge-enhanced memory evolution for remote sensing agents with systematic evaluation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32899–32915, 2026.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, et al. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*, 2026.

Jinghui Xu, Boyi Shangguan, Mengke Zhu, Hao Liu, Junhuan Jiang, Guangjun He, Pengming Feng, Shichao Jin, Bin Liang, Yongzhe Chang, et al. Bridging perception and action: A lightweight multimodal meta-planner framework for robust earth observation agents. *arXiv preprint arXiv:2605.04777*, 2026.

Shidong Yang, Ziyu Ma, Tongwen Huang, Xucong Wang, Renda Li, Yiming Hu, Yong Wang, and Xiangxiang Chu. Skillforge: Evolving verifiable skills for reinforcement learning agents. *arXiv preprint arXiv:2608.24747*, 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.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

Bo Yu, Cheng Yang, Dongyang Hou, Chengfu Liu, Jiayao Liu, Chi Wang, Zhiming Zhang, Haifeng Li, and Wentao Yang. Geoagentbench: A dynamic execution benchmark for tool-augmented agents in spatial analysis. *arXiv preprint arXiv:2604.13888*, 2026.

Zhiwei Zhang, Yudi Lin, Nikki Lijing Kuang, Linlin Wu, Xiaomin Li, Songtao Liu, and Fenglong Ma. Co-evolving skill generation and policy optimization. *arXiv preprint arXiv:2606.08755*, 2026.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

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.

Appendix at a Glance

A Data Construction and Training Protocol p. 13
Learning, selection, and evaluation roles; fixed allocations, optimization, and baseline settings.
B GeoState and GeoVerify p. 19
Verified dependencies, comparison objects, completion metrics, and independent annotation audit.
C Full GDL-GRPO Formulation p. 24
Credit semantics and event tables, opportunity normalization, and exact failure/token-update rules.
D GeoSkill Specification, Evaluation, and Evolution p. 28
Typed and concrete specifications, verified admission, matched effects, selection, and maintenance.
E Extended Experimental Analyses p. 33
Operational ablations, dependency closure, matched interventions, robustness, and transfer evidence.
F EO Execution Cases p. 43
Illustrative GIS, spectral-preview, and bounding-box execution traces.

A Data Construction and Training Protocol

OEA-TRAIN supplies cold-start interface alignment and on-policy learning of verified credit and skills. Figure 5 maps the data populations to their learning and evaluation roles.

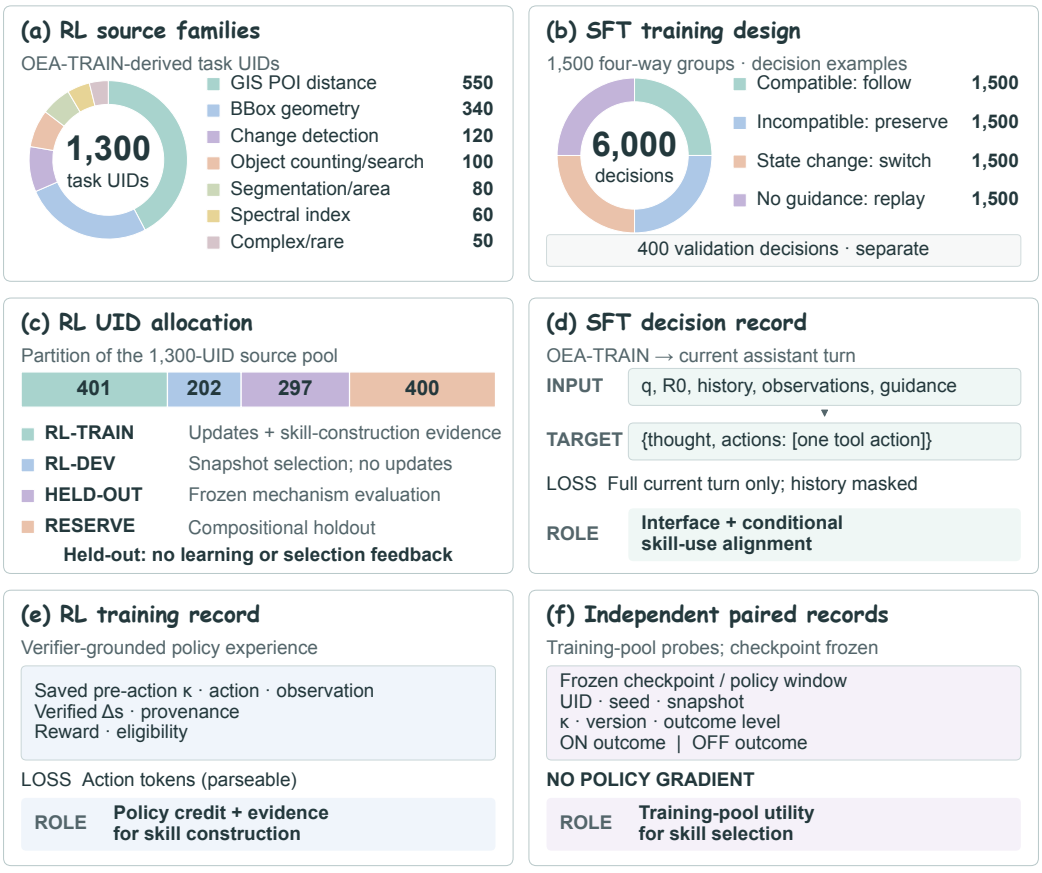

Figure 5: **Data construction and learning records.** (a) RL-source task UIDs; (b) SFT training decisions; (c) RL UID allocation. (d–f) Schematic fields, supervision targets, and downstream uses for the three record types. The two rings describe separate populations. RL action-token loss masks apply to parseable responses; malformed-response action spans follow Sec. C.3.

A.1 COLD-START SOURCE

Cold-start reconstruction uses existing EO queries, public image/GeoTIFF/GIS resources, official tool schemas, historical actions, real observations, subsequent valid actions, and termination or artifact handoffs. The resulting Tool-and-Skill-Use demonstrations initialize unadapted Qwen3.5-9B as π_{θ_0} before grounded RL; neither stage collects new EO raw data.

A.2 DECISION-LEVEL CONSTRUCTION AND SUPERVISION

A source trajectory supplies decision-level examples

$$x_t^{\mathrm{SFT}} = (q, R_0, h_t, o_{<t}, c_t) \longrightarrow a_t^{\star}. \tag{5}$$

Here q is the task, R_0 the public initial resources, h_t the preceding action–observation history, $o_{<t}$ its observation context, and c_t optional procedural guidance. The SFT superscript distinguishes these examples from verified RL records. The target $a_t^{\star}$ is the current valid assistant decision:

```
{"thought": "...",
 "actions": [{"name": "ToolName", "arguments": {...}}]}
```

The envelope contains exactly these two keys and one schema-valid action. Image, GeoPackage, and layer references resolve to established public bindings; Terminate requires readiness or grounded endpoint support. The thought is deterministically reconstructed from visible interaction/state conditions and skill applicability, without copying historical hidden reasoning.

SFT supervises the complete current two-key response, masking the system prompt, user query, historical assistant turns, tool observations, Skill text, and padding. For parseable responses, GDL-GRPO assigns loss to the current action-token span; malformed responses follow the support-recovery rule in Sec. C.3.

A.3 FOUR-WAY SKILL-CONSUMPTION GROUPS

The corpus contains 6,000 training and 400 validation decisions. Its 1,500 training groups each instantiate the four conditions in Figure 5(b), aligning conditional procedural use rather than memorizing the final bank.

- **Relevant guidance:** follow a compatible procedure.
- **Irrelevant guidance:** ignore it while preserving the valid action.
- **State change:** switch to the valid tool or stage.
- **No guidance:** act correctly without a procedure.

The demonstrations include GetAreaBoundary → AddPoisLayer, AddPoisLayer → ComputeDistance, TextToBbox → Calculator, Calculator → Terminate, AddIndexLayer → ComputeIndexChange, and ComputeIndexChange → Terminate handoffs.

A.4 CONSTRUCTION AUDIT

Demonstrations use source trajectories, public tool contracts, and deterministic compact rules. Cold-start construction makes no teacher API calls and does not read OEA-TEST or external or evolved GeoSkillBank entries. Teacher synthesis begins during RL from verified training evidence.

A.5 COLD-START COMPLIANCE DIAGNOSTICS

Cold start establishes the executable interface for grounded policy learning and skill use (Table 5).

A.6 RL SOURCE POOL

The 1,300-UID source pool (Fig. 5(a)) contains 550 GIS / POI, 340 BBox geometry, 120 Change detection, 100 Object counting / search, 80 Segmentation / area, 60 Spectral / index, and 50 Complex / rare UIDs. Their HELD-OUT counts are 126/78/27/23/18/14/11, respectively, totalling 297 (Table 21). The empirical family distribution is skewed rather than balanced.

Table 5: Cold-start interaction diagnostics (%). Columns measure interface compliance and readiness, separately from final OEA task performance.

Condition	Strict JSON	Exactly one action	DAG readiness	Grounded endpoint	Premature Terminate
Raw	66.2	51.8	1.8	5.4	8.9
Cold-start	100.0	100.0	25.0	19.6	0.0

A.7 Frozen splits and evaluation roles

We separate learning, selection, mechanism evaluation, and compositional generalization through four frozen UID-disjoint partitions:

$$401_{\text{RL-TRAIN}} + 202_{\text{RL-DEV}} + 297_{\text{HELD-OUT}} + 400_{\text{RESERVE}} = 1{,}300.$$

Split	Role and permitted use
RL-TRAIN	Learning: policy updates, teacher-guided skill construction, and training-time utility fitting.
RL-DEV	Selection: no-update model/bank validation and learning-dynamics measurements.
HELD-OUT	Mechanism evaluation with frozen policies and banks.
RESERVE	Compositional generalization; source of the 150-task holdout.

RESERVE contains the 400 UIDs shown in Fig. 5(c). Independent OEA-TEST contains 1,169 tasks, evaluated after checkpoint and final-bank freezing.

Partition boundaries. RL-DEV, HELD-OUT, and RESERVE, including source trajectories and derived records, are excluded from SFT training/validation, teacher-guided skill construction, candidate construction, training-utility fitting, and policy gradients. Training-utility fitting uses RL-TRAIN only. The independent validation distribution uses RL-DEV-BANK and is not merged with the training posterior. HELD-OUT and RESERVE provide no model or bank selection feedback. The composition holdout additionally excludes its task instances, complete source trajectories, designated critical edge pairs, and three-node dependency motifs from SFT, policy training, skill synthesis, verified admission, and training-time utility probes. Individual tools, semantic roles, and requirement-node types may occur in training; composition evaluation uses frozen policies and banks.

A.8 Training records and no-update probe records

Training records contain the public interaction, task endpoint contract, task DAG, verified state deltas, rewards, failure attribution, skill delivery information, the exact generated response, rollout-time token offsets, action-token loss masks, and behavior log-probabilities. The policy sees the OEA system and tool contract, the two-key/one-action contract, the public task, public initial resources, the visible action and observation history, the current Task carrier, the current State carrier, and the latest real observation. It does not see the full GeoState serialization, the requirement graph, the saved κ key, the readiness vector, reward components, risk labels, or verifier annotations. A State carrier replaces the current State-guidance block at an applicable frontier; it does not append another block, and the previous State guidance is not kept. When context exceeds 8,192 tokens, the oldest completed action–observation exchanges are deleted first, while the contract, query, current public resources, both carriers, and the latest real observation are kept. If that protected content itself exceeds the limit, the step is unavailable: no tool runs, the policy receives no penalty, and $\omega = 0$.

Training records supply gradients and candidate-construction evidence. No-update probes restore saved snapshots at frozen policy checkpoints and record the candidate, policy window, and paired outcome and risk differences. Posterior utility uses RL-TRAIN records. Independent Selection-safe validation uses RL-DEV-BANK and does not add those pairs to training support or the training posterior (Sec. D).

A.9 MATCHED TWO-PHASE POLICY-TRAINING ROLLOUT BUDGET

Both the policy-only GDL arm θ_P and the co-evolved arm θ_C start from the same SFT checkpoint and receive 401 Phase-I groups plus 140 Phase-II groups, with eight policy trajectories per group:

$$(401 + 140) \times 8 = 4{,}328.$$

Both arms share the GDL reward, optimizer configuration, rollout budget, checkpoint-selection rule, and jointly frozen Phase-II task schedule; θ_P keeps skills disabled throughout RL. Phase I emphasizes coverage; Phase II provides balanced policy–skill co-adaptation in the co-evolved arm, with the policy-only arm following the same tasks.

The budget counts policy-training rollouts, including groups that skip updates because they lack eligible advantages. Action lengths, tool calls, and realized optimizer steps may differ. Additional no-update admission and utility probes are tracked separately.

Fixed treatment allocation. Each group freezes the behavior policy π_{θ_g} and bank B_g. In both phases, its eight co-evolution rollouts preassign four OFF slots (no guidance), three CURRENT slots (active-bank guidance), and one NEW slot (applicable admitted versions acquiring support). The condition $c_{g,i} \in \{\mathrm{OFF}, \mathrm{CURRENT}, \mathrm{NEW}\}$ is assigned before outcomes. An empty applicable pool falls back through NEW $\to$ CURRENT $\to$ OFF or CURRENT $\to$ OFF; NEW remains capped at one slot. The policy-only arm uses OFF for all eight.

When all applicable pools are available, the planned NEW fraction is $1/8 = 0.125$ overall and $1/(3+1) = 0.25$ among ON rollouts. Fallback changes realized exposure: CURRENT vacancies become OFF and never create extra NEW slots. CURRENT requires an applicable Active, Retained Active, or Selection-safe version. Experimental and TrainBank versions remain eligible only for NEW, which selects the least-supported applicable admitted tier and breaks ties uniformly with the frozen RNG before observing outcomes. HOLD and disabled versions enter neither pool. Table 17 links support states to delivery eligibility.

Two-timescale schedule. Policy updates and utility evaluation use separate clocks. A nonzero update changes the live policy immediately; a skipped update leaves its parameters unchanged while the scheduled group cursor advances. Neither event triggers an unscheduled utility evaluation. The ten prespecified macro-boundaries are

$$(b_1, \ldots, b_{10}) = (44, 83, 163, 243, 323, 401, 441, 481, 521, 541).$$

Phase I contains the first six windows, including the boundary at group 401; Phase II contains the remaining four. At boundary b_w, the controller:

1. Completes the boundary group's rollout and policy-update step, then saves the post-group anchor checkpoint $\bar{\theta}_w$.
2. Fixes candidate identities and eligible snapshots after candidate construction, compilation, and independent admission as applicable.
3. Holds the policy and bank fixed while executing the no-update matched continuations, then checks paired outcomes and estimates posterior Π_w.
4. Applies the support and safety gates and commits bank $\bar{B}_w$. The next rollout group, not the completed boundary group, first uses this bank and posterior.

For $1 \le w < 10$, every group $b_w < g \le b_{w+1}$ uses $B_g = \bar{B}_w$ and selection posterior Π_w. Intermediate policy updates do not change these sealed objects; CURRENT samples from the most recent anchor-policy posterior, while NEW follows the reserved least-supported-tier allocation. Earlier observations retain provenance and cumulative support but do not enter the new anchor's utility posterior. The checkpoint saved after group 541 is eligible for final snapshot selection but supplies no subsequent training interval.

Delivery before bank activation. Before the first boundary, both skill pools are empty and all eight rollouts are OFF. Until a candidate meets the Active support gates, an admitted candidate may use the reserved NEW slot; vacant CURRENT slots become OFF, never extra NEW slots. The primary run first promoted an Active skill after the second utility window, so CURRENT guidance began in the following interval. It applied only to tasks with a suitable Active skill. Later windows prioritize previously unused task UIDs when accumulating distinct-UID support (Table 17).

Table 6: Cold-start SFT configuration for checkpoint θ_0. Composition counts refer to training decisions.

Setting	Evaluated value
Backbone	Qwen3.5-9B
Data	6,000 training / 400 validation decisions
Four-way composition	1,500 relevant-follow; 1,500 irrelevant-preserve; 1,500 counterfactual-switch; 1,500 no-skill replay
Context	8,192 tokens
LoRA	Rank 32, alpha 64, dropout 0
Precision / hardware	BF16; 2×A800-80GB with FSDP2
Micro / effective global batch	1 per GPU / 4; gradient accumulation 2
Optimizer	AdamW; LR 10^{-5}; betas $(0.9, 0.999)$; $\epsilon_{\text{opt}} = 10^{-8}$
Weight decay / gradient clip	0 / 1.0
Schedule	Constant learning rate, no warm-up or decay
Training length	1 epoch; 1,500 optimizer steps
Supervision	Current two-key assistant response (thought and action) only; history, tool receipts, Skill text, and padding masked

A.10 Fixed RL-DEV evaluation and snapshot selection

The 202 RL-DEV UIDs are split before training into RL-DEV-POLICY (138 UIDs) and RL-DEV-BANK (64 UIDs). Checkpoint selection uses GEC on the 138-UID policy panel, then VPR, a lower policy-invalid rate, and an earlier checkpoint. θ_P is the group-323 policy-only checkpoint; the selection rule chooses θ_C and $B^\star$ at group 521 while also evaluating group 541. The plotted learning curves report the recorded validation outcomes on the RL-DEV schedule. GEC measures grounded endpoint completion; VPR counts eligible decisions establishing a previously unmet task-required binding, predicate, readiness condition, or postcondition, excluding invalidation-only transitions and repeated support. Co-evolved OFF and Current Bank ON use the same checkpoint; at a macro-boundary, ON uses the bank committed for that anchor after its utility window. This condition uses only CURRENT-eligible versions, never Experimental or TrainBank versions. When no version is Active, ON supplies no guidance and is the same intervention as OFF under matched policy, seed, and execution conditions.

Prespecified RL-DEV-BANK validation supports the selection-safe gate without entering training support or the training posterior. The selected policy and bank freeze before HELD-OUT and OEA-TEST evaluation.

Figure 8 reports the checkpoint validation dynamics under these delivery rules; Table 33 records the scheduled checkpoint positions. Optimization reward, policy-invalid events, reward spread, KL, and clipping characterize training behavior separately from validation progress and completion.

A.11 Evaluated optimization and runtime configuration

Tables 6 and 7 specify cold-start θ_0 and RL, respectively, under the split rules in Sec. A.7. Malformed RL action support follows Sec. C.3.

Locally evaluated trained systems in the native and crossed tables report the prespecified primary run averaged over decoding seeds 11, 22, and 33. For GDL and TerraWeave, the primary training master seed is 0; three additional replications use master seeds 1, 2, and 3, paired across the two arms. Replication means and sample SDs exclude the primary run. Three additional ReSkill runs use the same decoding aggregation; all three methods are reported in Sec. E.11. Transfer tables report decoding-seed-averaged point differences without dispersion estimates.

The runtime contract registers 24 actions. Twenty-three are executed by tool-server workers; Terminate is the policy's control action and is not a worker-failure entry. The GIS chain is GetAreaBoundary, AddPoisLayer, ComputeDistance, DisplayOnMap. The spectral chain is AddIndexLayer, ComputeIndexChange, ShowIndexLayer. Table 8 summarizes the shared runtime; full schemas stay in the release artifact.

Table 7: GDL-GRPO and co-evolution configuration. Policy-rollout counts exclude no-update admission and utility probes. Independent runs execute sequentially with the listed hardware per run.

Setting	Evaluated value
Algorithm / initialization	GDL-GRPO / cold-start θ_0
Trainable parameters	Language LoRA: rank 32, alpha 64
Frozen components	Base model, vision encoder, reference policy
Phases / policy rollouts	401 / 140 groups; 8 per group; 4,328 total
Context / output / thought	8,192 / 1,024 / 64 tokens
Sampling	Temperature 0.7, top-p 0.95, top-k 0
Maximum turns	15
Optimizer	AdamW; LR 5×10^{-6}; betas $(0.9, 0.999)$; $\epsilon_{\text{opt}} = 10^{-8}$
Weight decay / schedule	0 / constant learning rate, no warm-up or decay
PPO clip / KL coefficient	$\epsilon_{\text{clip}} = 0.2$ / $\beta = 0.001$
Gradient clipping	1.0
Update rule	At most one optimizer step per group with nonzero eligible advantages
RL loss mask	Current action tokens; malformed support follows Sec. C.3
Training runs (GDL / TerraWeave)	Primary seed 0 plus three additional replications (seeds 1, 2, 3), executed sequentially
Decoding seeds	11, 22, 33 for each frozen evaluation checkpoint
Utility windows	10 prespecified macro-boundaries; Sec. A.9
Checkpoint selection	GEC on 138-UID RL-DEV-POLICY; ties by VPR, lower policy-invalid rate, earlier checkpoint. θ_P at group 323; $\theta_C/B^\star$ at group 521
Hardware per run	5×A800-80GB
Runtime allocation	1 updater; 2 persistent rollout workers; 1 boundary/probe worker; 1 tools/judge worker

Table 8: Runtime contract shared by policy rollouts and matched continuations.

Item	Frozen value
Registered actions	24
Tool-server workers	23
Control action	Terminate, not a worker failure
GIS chain	GetAreaBoundary → AddPoisLayer → ComputeDistance → DisplayOnMap
Spectral chain	AddIndexLayer → ComputeIndexChange → ShowIndexLayer

Qwen3-4B and Qwen3-4B-2507 abbreviate Qwen3-4B-Instruct-2507. At 4B scale, Qwen3-4B-Instruct-2507 first undergoes cold-start SFT; grounded RL starts from that checkpoint. The unadapted instruction model is the initial baseline, while the TerraWeave row includes alignment and RL. Frozen-bank transfer attaches $B^\star$ to independently trained policies under the common OEA tool contract, with target-policy parameters fixed.

A.12 EO-ADAPTED COMPARISON METHODS

Core policy baselines. Standard GRPO, GRPO+GeoVerify, and policy-only GDL share cold-start θ_0, ordered task and seed schedules, 541 groups of eight rollouts (4,328 trajectories), optimizer, action-token mask, OEA tool environment, skills-OFF condition, and the same RL-DEV checkpoint-selection rule. They differ in feedback and the unit of advantage assignment:

Condition	Training feedback	Advantage assignment
Standard GRPO	Native OEA trajectory score	One advantage per trajectory within a task group
GRPO+GeoVerify	Trajectory aggregation of GeoVerify events	One advantage per trajectory within a task group
GDL-GRPO	GeoVerify decision-local opportunity reward	One advantage per decision within its semantic κ group

For $b \in \{\text{native}, \text{GV}\}$, let $\mathcal{G}_g(q)$ contain trajectories sampled for task q in group g. The two trajectory baselines use

$$A_i^{(b)} = \frac{R_i^{(b)} - \text{mean}_{j \in \mathcal{G}_g(q_i)} R_j^{(b)}}{\text{std}_{j \in \mathcal{G}_g(q_i)} R_j^{(b)} + \epsilon}.$$

The resulting scalar is applied to all loss-bearing action spans of trajectory i. Standard GRPO uses the native OEA trajectory score R_i^{native}; it has no GeoState process credit, valid-negative credit, semantic grouping, or skill guidance. GRPO+GeoVerify instead uses the same verified event ledger as GDL, aggregated into the accounting return in Eq. (19):

$$R_i^{\text{GV}} = R_i^{\text{acct}} = \min\left\{0.10, \sum_{t=1}^{T_i} \omega_{i,t} c_{i,t}^{\text{loc}}\right\} + \sum_{t=1}^{T_i} \omega_{i,t}\left(c_{i,t}^{\text{VN}} - p_{i,t}\right) + \sum_{t=1}^{T_i} c_{i,t}^{\text{end}}.$$

The trajectory process cap is applied once after rollout completion, not as a sequential per-step budget. GDL instead normalizes $r_{i,t}^{\text{opp}}$ within admitted $\mathcal{C}_\kappa^{(g)}$ sets, with the eligibility and failure-adjustment rules in Sec. C.2. The first comparison changes verification feedback; the second evaluates GDL's local-credit design against trajectory-level aggregation under shared events. The full-system ablation in Sec. E.2 tests the comparison key.

Other adapted methods and accounting. The four adaptations in Table 9 share Qwen3.5-9B, cold-start θ_0, language LoRA rank 32/alpha 64, UIDs, seeds, and 541 groups of eight policy rollouts (4,328 trajectories). They use the frozen OEA runtime and rollout, decoding, optimizer, action-loss, and selection controls in Table 7. HELD-OUT and OEA-TEST remain evaluation-only. All methods in main Table 2 receive the same 4,328 policy-training trajectory allowance. D2Skill's baseline/guided comparisons and ReSkill's version tests consume that allowance. In contrast, TerraWeave's independent admission and matched-utility probes are additional no-update executions outside it. The shared teacher request cap is an allowance, not matched realized usage. Thus these comparisons match policy-training rollouts, not total tool calls, generated tokens, teacher usage, or compute; execution accounting is reported separately in Sec. E.11. The policy-input projection remains the public query, resources, interaction history, and assigned guidance (Sec. A.8). D2Skill retrieves from task text or visible observations/history; ReSkill uses public tool arguments and observations. Their GeoVerify reward feedback does not add the verifier-side frontier or readiness fields to these retrieval inputs; their adapted representations are specified below.

B GeoState and GeoVerify

B.1 State objects and provenance

GeoState $s_t = (R_t, z_t, P_t, \xi_t)$ provides a persistent dependency reference for policy credit and procedural evaluation. GeoVerify instantiates and updates it from observations. Table 10 defines the resource, predicate, and postcondition records together with their certifying provenance.

The action and returned observation produce $\Delta s_t^{\text{ver}} = \text{Verify}_q(s_t, a_t, o_{t+1})$, establishing bindings, predicates, or postconditions and invalidating those with stale resource versions or provenance. Positive verified progress requires a previously unmet task condition. Repeated support adds provenance; invalidation-only transitions update the state without VPR or positive process credit. Here Δs_t^{ver} contains the logical resource, predicate, and postcondition changes. The accompanying provenance change $\Delta\xi_t$ is retained as a separate field in the execution record (Sec. C.1); additional supporting observations need not establish a new logical atom.

B.2 Events and task dependencies

Task endpoint contracts define the requested answer or artifact, resource roles, and observation-supported closure conditions. Their graphs are instantiated from a frozen task-contract library $\mathcal{T}_{\text{family}}$, represented as semantic requirement templates with executable verification conditions:

$$G_q = \text{Instantiate}(\mathcal{T}_{\text{family}}, q, \mathcal{M}_q, R_0, \mathcal{A}). \tag{6}$$

Table 9: EO baseline adaptations. Rows identify source papers and implementation provenance; columns specify retained mechanisms, feedback, matching, and settings. Teacher budgets are request allowances.

Method / source	Retained mechanism and EO substitution	Method-specific settings, selection, and resources
GiGPO + GeoVerify NeurIPS 2025; official implementation	Episode macro advantage plus weighted step advantage, discounted step returns, and critic-free clipped updates. OEA public observations, normalized tool results, resource aliases/types, and recent context replace ALFWorld/WebShop states; transient resource identifiers are canonicalized. Main-run anchors match exactly, without frontier/target/readiness fields. Episode advantage uses the shared GeoVerify trajectory return; step returns use its decision-level event rewards. Environment failures and controllable penalties use the shared rules.	$\gamma = 0.95$, $\omega = 1.0$, support ≥ 2; similarity matching OFF (optional threshold 0.95). RL-DEV sensitivity: $\gamma \in \{0.90, 0.95, 1.00\}$, $\omega \in \{0.5, 1, 2\}$. No teacher calls or skill probes. Qwen3.5-9B/LoRA and the shared EO optimizer and schedule replace the original Qwen2.5/full-model environment setup; macro/micro credit remains intact.
SALT + GeoVerify Findings of EACL 2026; paper-faithful local reimplementation of Algorithms 1–2	One trajectory graph per task/group. States contain recent action–observation pairs; identical (before-state, action, after-state) edges merge and receive mean trajectory advantage; unique edges retain their trajectory advantage. OEA structured history and normalized JSON/aliases/file IDs replace environment text encoding. A shared scalar GeoVerify trajectory score supplies group-relative advantages; SALT redistributes them through the graph, without direct local-reward credit or dependency-key matching.	History $h = 3$, shared-edge support ≥ 2, exact structured matching; embedding matching OFF. RL-DEV sensitivity $h \in \{1, 3, 5\}$. No teacher calls or skill probes. Model/environment/optimizer substitutions use the shared controls; graph construction and advantage redistribution retain the paper's mechanism.
D2Skill official implementation	Task and step banks, reset/observation retrieval, baseline/guided A/B rollouts, hindsight trajectory-gap utility, EMA/UCB retrieval, dynamic generation and pruning, and intrinsic utility reward. Free-text title/principle/applicability/ retrieval fields remain. Queries use task text or visible observation/history; Qwen3-Embedding-0.6B retrieves guidance. Policy base reward is the shared GeoVerify trajectory return. Utility compares current-policy/task-group outcomes, without typed compilation or saved-state utility.	A/B baseline fraction 0.5; top-1 Task and step retrieval; similarity 0.7; EMA 0.5 for both banks; retrieval pool 10, $\alpha = 0.1$, UCB $c = 0.05$; intrinsic coefficient 1.0. Bank limits 300/300; update success threshold 0.5, up to 10 trajectories/update, 10 protected recent updates. Official WebShop configuration; DEV checks top-k $\{1, 3\}$ and thresholds $\{0.6, 0.7, 0.8\}$. Shared o3-mini teacher, at most 300 requests; A/B tests use in-budget policy rollouts, no extra saved-state probes. Policy and both banks freeze together.
ReSkill official implementation	Empty initial registry; experience reservoir; assertion CRUD/diagnosis; ADD/MODIFY/DELETE bundles; trigger validation; within-group old/new version testing; Beta–Binomial Thompson sampling with adaptive posterior discounting; atomic bundle acceptance and pruning. EO tool names, public arguments and observations replace ALFWorld/Search trigger vocabulary. GRPO uses the shared GeoVerify trajectory return; version success is strict grounded episode completion. Comparisons share task/policy groups, not exact execution snapshots.	Testing steps 5; at least 50 episodes/version; Beta(1,1) priors; 10,000 Thompson samples; assignment clamp $[0.15, 0.85]$; dynamic weighting. Reservoir 100–200; active skills ≤ 8; trigger rate ≥ 0.5; retries 3; skill text ≤ 500 characters; creation groups 30; success ratio 0.3; history 5. Official defaults; TRAIN/DEV checks parser/tool compatibility. Shared o3-mini replaces Claude; at most 300 requests, in-budget version tests, no extra saved-state probes. Registry/posterior freeze with the policy.

Table 10: Logical GeoState schema. A resource or condition is supported by a real observation rather than by its occurrence in a policy response.

Object	Contents and role
R_t	Typed raster, vector, GIS-layer, and artifact records; public aliases, resource bindings, and producing-action lineage.
z_t	Verified task predicates, including entity/region relations, temporal or spectral identity, and value/unit conditions.
P_t	Preconditions for downstream consumption, valid artifact postconditions, and requested endpoint closure.
ξ_t	Map from each established resource or condition to its supporting observation set in $O_{1:t}$; receipt and lineage links remain attached.
G_q	Typed requirement nodes with identifiers, categories, parent dependencies, and verification requirements.

Here $\mathcal{M}_q$ is public metadata, R_0 the initial public resources, and $\mathcal{A}$ the tool contracts. Public task information defines the requirements, each admitting multiple compatible producers; observations establish their satisfaction. Templates specify what to establish, whereas GeoSkills specify when and how.

Template construction and freezing. The library contains seven family roots and 24 executable leaf templates. Authors define the semantic nodes, dependency edges, resource roles, legal producers, satisfaction conditions, invalidation rules, readiness rules, and endpoint contracts. Two authors check the templates independently and resolve disagreements jointly. Only RL-TRAIN trajectories and their permitted historical training records are used to calibrate producer mappings, observation grammars, resource roles, value/unit checks, readiness, and invalidation. Calibration and coverage tests finish before template freezing, SFT construction, and all policy training. RL-DEV, HELD-OUT, RESERVE, and OEA-TEST do not inform template modification. Template, verifier-rule, and tool-contract versions are fixed across experimental conditions.

Family	Leaves	Requirement archetypes
BBox geometry	4	Localization; attribute/direction; geometry/value; annotated artifact
Change detection	3	Qualitative change; quantified change; change artifact
Complex / rare	3	External fact and calculation; multi-source reasoning; mixed/rare artifact
GIS / POI	4	Boundary/POI; distance; overlay; distance-and-overlay closure
Object counting / search	3	Direct visual count; conditional/region count; search-based fact count
Segmentation / area	4	Mask/pixels; object-wise count; physical area; ratio/derived geometry
Spectral / index	3	Single index; temporal change; preview/artifact set
Total	24	Seven family roots

Deterministic instantiation. A parser extracts entity, location, date, unit, and artifact slots from the public query and task metadata. Initial resources bind semantic roles; frozen tool contracts constrain legal producers and preconditions. The parser selects one or more leaf templates, composes their nodes, and topologically sorts the resulting graph. Final requirement semantics are author-defined and runtime instantiation makes no LLM call. Parsing failures fail closed rather than requesting an LLM-generated replacement graph.

A valid negative is a usable observation against a candidate relation that changes the reachable plan. A controllable failure is attributable to the policy's action, arguments, resource choice, or stop decision. An unavailable environment supplies no valid policy-learning opportunity. These event types remain distinct in the execution records.

B.3 DEPENDENCY COMPARISON OPPORTUNITIES

Let $\mathrm{Sat}_q(s_t)$ be the requirement nodes currently verified from the resource bindings R_t, predicates z_t, and postconditions P_t, with observation provenance ξ_t. Then

$$\mathrm{frontier}(s_t) = \{a \in G_q \setminus \mathrm{Sat}_q(s_t) : \mathrm{Parents}(a) \subseteq \mathrm{Sat}_q(s_t)\}, \tag{7}$$

$$\mathrm{target}(s_t) = \mathrm{first}_{\prec_q} \mathrm{frontier}(s_t). \tag{8}$$

For a nonempty frontier F_t, $\prec_q$ fixes topological and semantic-role precedence and selects $r_t^\star$. Readiness is recorded for every frontier requirement, not only this active target:

$$\rho_t^F = \big\{(r, \rho_t(r)) : r \in F_t\big\}, \qquad \kappa_t = (q, F_t, r_t^\star, \rho_t^F).$$

For each r, $\rho_t(r)$ records whether the requirement is ready; whether each required role is bound; its resource type and canonical semantic signature; non-emptiness and current-version validity; and other execution preconditions. This structure is saved before action generation and remains fixed for comparison within the training group.

The active target serves as the reference for local credit, plan-changing negative evidence, and candidate or State Skill applicability under the fixed priority. It does not exclude actions that establish other nodes in F_t. Two states with the same active target but different execution conditions for another frontier node require different comparison keys.

Exact identity and semantic canonicalization. Invalidation checks a resource's exact binding, producing action and arguments, supporting observation, content identity, and runtime version. Cross-trajectory comparison instead uses

$$\begin{aligned} c_q(h) = (&\text{task role}, \text{resource type}, \text{entity/filter}, \\ &\text{spatio-temporal support}, \text{CRS compatibility}, \\ &\text{source-data version}, \text{lineage signature}). \end{aligned}$$

Exact identity drives invalidation; semantic identity drives comparison. Differently named hospital layers can match when these semantic fields agree; identically named boundaries covering different cities cannot. Source-data versions and lineage remain comparison fields regardless of resource aliases.

For comparison, we sort frontier requirements by the frozen requirement order and resource roles by name. We omit temporary storage locations, run-specific identifiers, and private aliases, but retain semantic resource signatures, source versions, and lineage. No action, subsequent observation, reward, future outcome, or trajectory process-budget remainder enters the saved key.

Frontier membership differs from execution readiness: a requirement can be reachable in the task graph even when its concrete execution handles are unbound. Any compatible producer, including a pre-existing resource, can satisfy the type, spatial, source-version, CRS, and provenance contracts. An action receives credit for a non-target requirement only when that node belongs to the saved F_t, its pre-action conditions are represented in ρ_t^F, and the observation establishes net-new verified support. Later observations update GeoState without relabeling the saved opportunity.

Executable intervention state. The saved execution snapshot $\zeta_t = (h_t^{\mathrm{vis}}, s_t, \mathcal{X}_t, \mathcal{M}_t)$ contains visible history h_t^{vis}, the executable contents and versions $\mathcal{X}_t$ referenced by R_t, and tool/environment state $\mathcal{M}_t$. Skill interventions restore separate copies of ζ_t with the same saved key and continuation seed, fixing the prefix, resource contents/versions, and environment. Realization is assessed against the candidate version's declared consequence.

Thus κ_t compares dependency opportunities, while ζ_t supports executable interventions. Equal keys need not imply identical environments or future values; local reward comparison (Sec. C.1) does not imply path-independent total trajectory reward.

GIS opportunity from an execution trace. The hospital–school task requests both POI layers, their minimum distance in kilometres, and an overlay containing both layers. Once the extent, boundary workspace, and both POI layers are verified, the distance relation and overlay are reachable but unmet. With these requirements denoted by r_4 and r_5, $F_t = \{r_4, r_5\}$ and the fixed priority selects

$r_t^\star = r_4$. Distance readiness records the current, nonempty hospital and school layers and compatible spatial references; overlay readiness records both layers and an available renderer. The saved key therefore includes both:

$$\kappa_t = \big(q, \{r_4, r_5\}, r_4, \{(r_4, \rho_t(r_4)), (r_5, \rho_t(r_5))\}\big).$$

A state with identical distance readiness but unavailable rendering would not share this key. An action may nevertheless establish either reachable requirement; the fixed priority is not a prescribed tool sequence.

The recorded `ComputeDistance` response gives 1,240 metres with source/target lineage. GeoVerify establishes the required distance atom, but the overlay remains unmet. The relation earns 0.02 process credit; target readiness was already true and endpoint readiness remains false. The finite-value, non-negativity, unit, and lineage checks support this one required relation rather than each receiving separate credit. Rendering the overlay subsequently establishes endpoint readiness, after which grounded termination can close the task. This execution trace shows how verified state grounds local credit.

B.4 Completion and repair

All mechanism variants use the same frozen GeoVerify configuration for evaluation. Ablations modify the learning or routing mechanism; the common state evaluator supplies comparable VSC, GEC, and FRR measurements. For task q, $\mathcal{A}_q^{\mathrm{req}}$ contains its required verified state atoms and grounded endpoint. Excluding the endpoint gives $\overline{\mathcal{A}}_q^{\mathrm{req}} = \mathcal{A}_q^{\mathrm{req}} \setminus \{\texttt{grounded_endpoint}\}$. Let $\mathcal{A}_{q,T}^{\mathrm{valid}}$ contain only currently valid, observation-supported atoms at termination. Then Verified State Completion is

$$\mathrm{VSC}(\mathcal{Q}) = \frac{100}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \frac{\sum_{a \in \overline{\mathcal{A}}_q^{\mathrm{req}}} \mathbf{1}[a \in \mathcal{A}_{q,T}^{\mathrm{valid}}]}{|\overline{\mathcal{A}}_q^{\mathrm{req}}|}. \tag{9}$$

VSC macro-averages non-endpoint completion across tasks. Invalidated atoms receive no credit; equivalent chains establishing the same currently valid atom receive equal credit. GEC separately measures endpoint completion:

$$\mathrm{GEC} = \frac{1}{297} \sum_{q \in \mathcal{Q}_{\mathrm{HELD}}} \mathbf{1}[\text{grounded requested endpoint at termination}]. \tag{10}$$

Here $|\mathcal{Q}_{\mathrm{HELD}}| = 297$; unsuccessful tasks stay in the denominator. GEC implies verified dependency closure, but prerequisite closure alone need not deliver the requested endpoint. For the common blocked-state panel $\mathcal{B}_{\mathrm{full}}$ of 591 repair opportunities, let $\mathrm{restored}(b)$ mean that later observations re-establish the unmet or violated predicate. The event-level definition is

$$\mathrm{FRR} = \frac{\sum_{b \in \mathcal{B}_{\mathrm{full}}} \mathbf{1}[\mathrm{restored}(b)]}{|\mathcal{B}_{\mathrm{full}}|}. \tag{11}$$

GEC and FRR are fractions reported as percentages (multiply by 100), with differences in percentage points (pp). FRR measures frontier restoration; repair-to-endpoint additionally requires downstream closure. VPR measures the fraction of eligible decisions establishing positive verified progress, rather than final state completeness.

Evaluation populations. Native scores use metric-specific eligibility within 1,169 OEA-TEST tasks (Sec. E.1). HELD-OUT Answer, dependency closure, and GEC use 297 tasks; VSC averages 297 non-endpoint completion ratios. $\mathcal{B}_{\mathrm{full}}$ FRR measures recoverability from 591 common blocked states (Sec. E.2); snapshot FRR and R2E use a separate 591-pair set from 207 UIDs (Sec. E.8). The depth VPR audit contains 12,667 eligible of 16,240 decisions, and matched L1/L2/L3 effects use 400/600/800 pairs. These outcomes retain their own statistical units even where populations overlap.

B.5 GeoVerify reliability

Independent EO annotations assess resource binding, required-atom detection, endpoint certification, and opportunity matching (Table 11). The transition audit contains 400 records stratified across

Table 11: **Independent validation of the shared state interface.** Counts use the audit units specified below. TP, FP, FN, and TN denote true positives, false positives, false negatives, and true negatives. Eligible-comparison coverage (12,667/16,240) is a separate decision-audit quantity, not an annotation accuracy.

Measure / audit unit	Counts	Score (%)
Resource-binding accuracy: 200 binding-bearing transitions	198 correct; 2 incorrect	99.0
Required-atom precision / recall: atom labels over 400 transitions	TP 394; FP 6; FN 10	98.5 / 97.5
Grounded-endpoint agreement: 150 closure records	147 agreements; 3 disagreements	98.0
Matched-opportunity precision / recall: 500 pairs	TP 245; FP 5; FN 5; TN 245	98.0 / 98.0

seven families and new-predicate, no-op, contradiction, and failure events. Of these, 200 contain a resource-binding instance; required-atom labels are pooled across all 400 transitions; one transition may contribute several atom labels. Endpoint certification uses 150 closure records. A separate opportunity-matching audit contains 500 pairs: 250 equivalent-opportunity pairs and 250 hard negatives. Matching precision and recall characterize this balanced audit population. Two annotators independently label method-blind and arm-blind records, with a third adjudicating disagreements. Pre-adjudication Cohen's κ is 0.94 for binding, 0.91 for requirements, 0.88 for endpoints, and 0.90 for matched opportunities. Comparison coverage, 12,667/16,240 (78.0%) in the separate depth audit, measures eligibility rather than annotation accuracy. VSC, GEC, and FRR measure downstream agent outcomes under this frozen verifier.

C Full GDL-GRPO Formulation

C.1 Reward semantics, events, and accounting

GDL-GRPO rewards new verified dependencies, plan-changing evidence, and grounded completion. Reward events define local credit; trajectory accounting applies a separate cap after rollout completion. Within rollout group g, records are

$$x_{i,t} = (s_{i,t}, \kappa_{i,t}, a_{i,t}, o_{i,t+1}, \Delta s^{\mathrm{ver}}_{i,t}, \Delta\xi_{i,t}, e_{i,t}), \qquad X^{\mathrm{train}}_g = \{(c_{g,i}, \{x_{i,t}\}_{t=1}^{T_i})\}_{i=1}^{n_g}. \tag{12}$$

Event label $e_{i,t}$ distinguishes progress, endpoint evidence, valid negatives, controllable failures, and environment unavailability (Table 13). The delivery condition $c_{g,i}$ is fixed for each rollout; state-specific carriers follow the routing rule. The condition is separate from reward. Usable records have $\omega_{i,t} = 1$; external environment failures are excluded before normalization and policy loss. Controllable failures, including malformed output, retain their penalties.

Opportunity reward. Process credit rewards net-new verified requirements and readiness transitions. The evaluated caps are $C_{\mathrm{dec}} = 0.06$ per decision and $C_{\mathrm{traj}} = 0.10$ for completed-rollout process accounting. Table 12 defines the events: $N^{\mathrm{new}}_{i,t}$ counts each semantic atom once across resource, predicate, relation/value/unit, and artifact categories; target and endpoint readiness each contribute on their first false-to-true transition.

$$c^{\mathrm{raw}}_{i,t} = 0.02 N^{\mathrm{new}}_{i,t} + 0.02\mathbf{1}[\Delta\mathrm{targetReady}] + 0.02\mathbf{1}[\Delta\mathrm{endpointReady}], \tag{13}$$

$$c^{\mathrm{loc}}_{i,t} = \min\{c^{\mathrm{raw}}_{i,t}, 0.06\}, \tag{14}$$

$$c^{\mathrm{VN}}_{i,t} = 0.005\,\mathbf{1}[\text{first independent plan-changing valid negative}], \tag{15}$$

$$c^{\mathrm{end}}_{i,t} = 1.00\,\mathbf{1}[t = t^{\mathrm{end}}_i \ \wedge\ \text{strict endpoint contract passes}], \tag{16}$$

$$r^{\mathrm{opp}}_{i,t} = \omega_{i,t}\left[c^{\mathrm{loc}}_{i,t} + c^{\mathrm{VN}}_{i,t} - p_{i,t}\right] + c^{\mathrm{end}}_{i,t}. \tag{17}$$

This is main-paper Eq. (3) with evaluated coefficients. Comparison equations use $r_{i,t}$ as shorthand for r^{opp}. Credited nodes must belong to the saved frontier with preconditions represented in $\rho^F_{i,t}$,

including for non-target nodes. Recording readiness across the frontier does not create a bonus for every frontier node: only the active-target and endpoint readiness transitions in Table 12 receive readiness credit. Three new atoms plus target readiness yield 0.08 raw credit, capped locally at 0.06.

Table 12: Evaluated process-event credit. The four atom categories share one deduplicated count; readiness transitions are separate. Every entry enters the 0.06 decision cap and 0.10 post-rollout process-accounting cap.

Process event	Credit	Required evidence
New required resource binding	+0.02/atom	Valid type, public binding, and receipt provenance
New required predicate	+0.02/atom	Previously unmet requirement becomes observation-verified
New required relation/value/unit	+0.02/atom	Entity, value, unit, or relation supported by a real observation
New required artifact postcondition	+0.02/atom	Artifact exists and passes type, content, and lineage checks
Target readiness false-to-true	+0.02	All execution prerequisites of the active target first become satisfied
Endpoint readiness false-to-true	+0.02	All evidence needed for a strict grounded endpoint first becomes available

Valid negatives and grounded endpoints. Valid-negative credit requires a successfully executed tool and a legal negative observation that excludes a target or branch and changes the subsequent legal plan. It is issued at most once per trajectory and independent evidence signature. Repeated signatures, unchanged plans/frontiers, and unavailable environments receive zero valid-negative credit.

Binary endpoint credit goes once to the establishing decision, normally a valid Terminate or a verified-artifact action that completes and ends the task. The strict contract requires verified answer value/entity/unit, present and valid required resources and artifacts, observation/receipt provenance, satisfied task postconditions, readiness, and no accepted false STOP. Partial completion, unsupported answers, missing provenance, and repeated endpoints/Terminate receive zero endpoint credit. Premature Terminate instead incurs the 0.06 penalty. Valid-negative and endpoint credit lie outside both process caps.

Table 13: Evaluated reward events and their local support. Process caps apply only to the first row.

Event	Supporting evidence	Contribution
Verified progress	New required binding, predicate, or postcondition	$c_{i,t}^{\mathrm{loc}}$ (local cap 0.06)
Valid negative	First independent negative observation that changes the plan	$c_{i,t}^{\mathrm{VN}} = 0.005$
Grounded endpoint	Observation-certified requested answer or artifact	$c_{i,t}^{\mathrm{end}} = 1.00$
Controllable failure	Failure attributable to the current action	$-p_{i,t}$
Unavailable environment	No usable environment evidence	$\omega_{i,t} = 0$

Invalidation and replay. Positive credit is keyed by the semantic requirement and is trajectory-local. A new receipt may add provenance without awarding the same semantic atom again. An upstream version change can invalidate established support. Re-establishing the atom in the same trajectory restores validity and does not grant that atom positive credit a second time. Destroy-and-restore therefore yields zero net progress, while the failure penalty remains. Reward history is not part of κ. A later visit to the same κ does not increase comparison support beyond that trajectory's earliest eligible decision.

Failure precedence. After excluding environment unavailability, the first applicable row of Table 14 assigns exactly one controllable-failure penalty. These mutually exclusive events have zero positive

process and valid-negative terms; Sec. C.2 bounds their advantages. Malformed output with no verified delta receives -0.06; a third repeated failure that also stagnates receives -0.05.

Table 14: Deterministic failure precedence. The first applicable row wins after environment quarantine; the magnitude is subtracted from reward.

Priority	Failure category	Penalty
1	Malformed output	0.06
2	Invented or unbound resource	0.06
3	Premature Terminate	0.06
4	Other policy-invalid action	0.05
5	Third repeated failure	0.05
6	Repetition / no verified delta	0.03
7	Stagnation	0.03

Post-rollout trajectory accounting. A newly established requirement predicate and a readiness transition can each contribute 0.02; together they contribute 0.04 before the local 0.06 cap. At rollout completion, the process total and accounting return are

$$C_i^{\text{proc}} = \sum_{t=1}^{T_i} \omega_{i,t} c_{i,t}^{\text{loc}}, \tag{18}$$

$$R_i^{\text{acct}} = \min\{0.10, C_i^{\text{proc}}\} + \sum_{t=1}^{T_i} \omega_{i,t} \left[c_{i,t}^{\text{VN}} - p_{i,t}\right] + \sum_{t=1}^{T_i} c_{i,t}^{\text{end}}. \tag{19}$$

The 0.10 cap applies once after completion, independently per rollout, and resets for the next rollout. It creates no sequential remaining-headroom variable: neither the capped total nor trajectory scaling enters r^{opp}, its opportunity mean/SD, or κ. Records separately retain raw components, the local cap, r^{opp}, process total, trajectory accounting, saved readiness, and advantages.

Events without an extra positive bonus. Credit tracks verified requirements, readiness, and closure. Tool/API success, Skill exposure/selection, recommendation adoption, changed actions, Teacher recommendations, repair labels, shorter trajectories, and ordinary JSON compliance have no separate bonus. Neither do repeated receipts, renewed support for satisfied predicates, or unverified resource creation.

C.2 Opportunity normalization and eligibility

Let $\mathcal{I}_g$ index the existing decisions in the current training group, and let $m_{i,t}^{\text{loss}}$ indicate nonempty recovered support with exactly aligned finite behavior log-probabilities (Sec. C.3). The comparison set and its mean are

$$\mathcal{C}_\kappa^{(g)} \subseteq \{(i,t) \in \mathcal{I}_g : \kappa_{i,t} = \kappa, \omega_{i,t} = m_{i,t}^{\text{loss}} = 1\}, \qquad \mu_\kappa^{(g)} = \frac{\sum_{(i,t)\in\mathcal{C}_\kappa^{(g)}} r_{i,t}^{\text{opp}}}{|\mathcal{C}_\kappa^{(g)}|}. \tag{20}$$

$\sigma_\kappa^{(g)}$ is the standard deviation of these same opportunity rewards. The normalized and failure-adjusted advantages are

$$A_{i,t}^{\text{GDL}} = \frac{r_{i,t}^{\text{opp}} - \mu_{\kappa_{i,t}}^{(g)}}{\sigma_{\kappa_{i,t}}^{(g)} + \epsilon}, \qquad \widetilde{A}_{i,t} = \begin{cases} \min(0, A_{i,t}^{\text{GDL}}), & \text{controllable failure,} \\ A_{i,t}^{\text{GDL}}, & \text{otherwise.} \end{cases} \tag{21}$$

Comparison support. Each trajectory contributes its earliest eligible decision per semantic κ; later revisits remain in execution/failure records without increasing support. Admission requires at least two eligible decisions, two distinct trajectory IDs, and two distinct *rollout seeds*. Rollout seeds identify sampled trajectories, distinct from the master seeds for the additional training runs in Table 7. Reward spread must satisfy

$$\max_{\mathcal{C}_\kappa^{(g)}} r^{\text{opp}} - \min_{\mathcal{C}_\kappa^{(g)}} r^{\text{opp}} > 10^{-8} \max\left\{1, \max_{\mathcal{C}_\kappa^{(g)}} |r^{\text{opp}}|\right\}. \tag{22}$$

Admission also requires a positive anchor: a loss-eligible successful decision with net-new verified requirement, readiness, or endpoint support, positive r^{opp}, and nonempty aligned action tokens. Valid-negative credit can enter the loss but cannot alone supply this anchor. Means and standard deviations use admitted current-group sets with their saved pre-action keys. Malformed or unsuccessful controllable decisions remain eligible when token support and aligned finite behavior log-probabilities are available.

Numerical settings. The normalization stabilizer is $\epsilon = 10^{-6}$. The scale-relative 10^{-8} spread tolerance rejects degenerate groups before normalization and is separate from the denominator. AdamW uses optimizer epsilon 10^{-8} (Tables 6–7), and PPO uses $\epsilon_{\mathrm{clip}} = 0.2$.

Training-time comparison support. In the primary policy-only GDL run, 366 of 541 scheduled rollout groups (67.7%) produced an optimizer update; 175 (32.3%) did not. Among admitted semantic comparison sets, mean/median support was 3.58/3 decisions, with interquartile range 2–5, 90th percentile 6, and 95th percentile 7. Each trajectory contributes at most once to a given set, so its support is bounded by the eight rollouts in the group.

The group-level diagnostics below use mutually exclusive gate precedence: no comparison set has sufficient distinct-rollout support; otherwise none passes the reward-spread gate; otherwise none has a grounded positive anchor. These are counts of zero-update groups, not individual rejected decisions.

Group outcome / first failed gate	Groups	Share of 541 (%)
Insufficient distinct-rollout support	78	14.4
Matched support, but no reward spread	61	11.3
Support and spread, but no grounded positive anchor	36	6.7
All zero-update groups	175	32.3
Groups producing an optimizer update	366	67.7

Percentages are rounded independently. No coarser fallback grouping or additional rollouts replace inadmissible comparisons. These statistics concern this primary GDL training run only; the 78.0% eligibility coverage in Table 22 concerns a separate common-snapshot evaluation, not training update coverage or the fraction of all decisions receiving loss.

Algorithm 1: GDL-GRPO update from saved dependency opportunities

1. Fix behavior policy π_{θ_g} and guidance assignments for eight rollouts. Before each action, save GeoState and $\kappa_{i,t}$.
2. Execute the generated decision, retain its observation or attributed failure, and update the verified state without relabeling the saved key.
3. Compute locally capped process credit and $r^{\mathrm{opp}}_{i,t}$. Recover current token support from the exact response and rollout offsets. Exclude external environment failures, empty support, and records lacking exactly aligned finite behavior log-probabilities before comparison.
4. Form current-group $\mathcal{C}^{(g)}_{\kappa}$ sets. Admit sets with the trajectory/rollout-seed support, earliest-eligible-contribution, relative-spread, and positive-anchor gates in Sec. C.2; normalize r^{opp} and cap controllable-failure advantages at zero.
5. Use the recovered token indices and aligned behavior probabilities. A zero mask contributes zero loss without evaluating an empty token mean. Average the clipped objective over trainable tokens within decisions, then over eligible decisions; add reference regularization.
6. Take at most one optimizer step if nonzero eligible advantages and trainable spans exist; otherwise set $\theta_{g+1} = \theta_g$.

Separately, each completed rollout receives its 0.10 process cap for trajectory accounting. This accounting does not alter Step 4 rewards or normalization.

C.3 Action spans and policy updates

Let $y_{i,t}$ be the current response and $K_{i,t}$ its recovered action-token indices. Parseable responses train on structured action tokens, with thought and other context outside the direct RL target. Current and behavior policies share the actual prefix $H_{i,t,k}$ of token $y_{i,t,k}$: visible history, assigned guidance, and preceding generated tokens.

$$\varpi_{i,t,k} = \frac{\pi_\theta(y_{i,t,k} \mid H_{i,t,k})}{\pi_{\theta_g}(y_{i,t,k} \mid H_{i,t,k})}. \tag{23}$$

This token ratio ϖ is distinct from per-requirement readiness $\rho_t(r)$ and its frontier-wide collection ρ_t^F.

Malformed output and empty masks. Recovery uses the exact assistant response and rollout-time token offsets, without repairing or retokenizing the output. $K_{i,t}$ may contain several recovered regions. An EOS-only response uses the sampled EOS token. Otherwise, support is recovered in the following order:

1. Use the value span of a strictly parsed top-level `actions` field.
2. If strict parsing fails, a string-aware lexer recovers every top-level or shallowest `actions` region. An unclosed container extends to the final generated content token.
3. If no `actions` key exists, recover the smallest object containing action-like `name` and `arguments` fields.
4. Attribute a schema-valid root object that omits `actions` to its closing decision boundary and sampled EOS.
5. For a max-length truncation reaching this fallback, use the final eight generated content tokens.
6. Only when no structured region exists, use the complete current assistant response as an envelope-level fallback.

Parseable actions retain an action-only mask; malformed envelope fallbacks can assign negative credit elsewhere in the current response. History, tool receipts, and supplied Skill text remain context.

Trainable support and update. Nonempty recovered support with exactly aligned finite behavior log-probabilities sets $m_{i,t}^{\text{loss}} = 1$. Otherwise $K_{i,t}$ is empty and $m_{i,t}^{\text{loss}} = 0$: the record is excluded from normalization, has zero loss weight, and retains its recorded failure penalty. The decision loss is

$$\ell_{i,t} = \begin{cases} -\dfrac{1}{|K_{i,t}|} \displaystyle\sum_{k \in K_{i,t}} \min\Big(\varpi_{i,t,k}\widetilde{A}_{i,t}, \text{clip}(\varpi_{i,t,k}, 1-\epsilon_{\text{clip}}, 1+\epsilon_{\text{clip}})\widetilde{A}_{i,t}\Big), & m_{i,t}^{\text{loss}} = 1, \\ 0, & m_{i,t}^{\text{loss}} = 0. \end{cases} \tag{24}$$

The empty-span branch precedes the token mean, avoiding evaluation of $1/|K_{i,t}|$ when $|K_{i,t}| = 0$. Let $\mathcal{E}_{\text{elig}}$ contain current-group decisions that pass the comparison-admission gates in Sec. C.2 and the token-support check $m_{i,t}^{\text{loss}} = 1$. For a nonempty eligible set, the group objective is

$$\mathcal{L}_{\text{GDL}}(\theta) = \frac{1}{|\mathcal{E}_{\text{elig}}|} \sum_{(i,t) \in \mathcal{E}_{\text{elig}}} \ell_{i,t} + \beta \mathcal{D}_{\text{KL}}. \tag{25}$$

Here $\mathcal{D}_{\text{KL}}$ is regularization against the fixed reference policy, with $\beta = 0.001$ (Table 7). Algorithm 1 applies Eq. (25) only when eligible trainable spans and nonzero eligible advantages exist. Otherwise the entire optimizer step, including reference regularization, is skipped and $\theta_{g+1} = \theta_g$.

D GeoSkill Specification, Evaluation, and Evolution

A skill moves from a typed specification through compilation and verified admission to matched evaluation, supported selection, and version maintenance.

D.1 TYPED SPECIFICATION

A GeoSkill declares its applicability and observable consequence. Its immutable version $v = (\gamma, \ell, \chi; \texttt{when}, \texttt{act}, \texttt{verify}, \texttt{repair}, \texttt{stop})$ separates delivery scope from consequence: Task skills guide a reusable deliverable from initialization, while State skills repair or advance a missing requirement at a verified frontier. L1 evaluates action/resource transitions, L2 semantic value/unit consequences, and L3 grounded endpoint/artifact closure. The evidence certificate χ links the procedure to provenance-backed records and the grounded evidence needed to instantiate and verify it. Table 15 maps these fields to executable checks.

Table 15: SkillSpec fields and their executable interpretation.

Field	Interpretation
γ, ℓ, χ	Task/State delivery scope, L1–L3 consequence, and supporting evidence.
`when`	Family, phase, frontier, target predicate, and required resources.
`act`	Public action/tool, typed argument roles, and producer/consumer relation.
`verify`	Required observation or postcondition certifying the consequence.
`repair`	Supported recovery relation at an unmet or violated frontier.
`stop`	Evidence-dependent closure condition.

Concrete specifications. The following State and Task examples instantiate the typed fields above; the evidence certificate and routing metadata follow the same schema.

Field	**Boundary binding** (State skill)	**POI-overlay delivery** (Task skill)
Purpose	Advance one missing requirement.	Complete the requested artifact.
`WHEN`	Extent observed; boundary package absent.	POI-overlay request; output incomplete.
`ACT`	GetAreaBoundary with the observed tuple.	Bind extent, add POIs, render on GeoTIFF.
`VERIFY`	GPKG and boundary table persist.	Resource handoffs and source-linked output.
`REPAIR`	On format failure, normalize and retry.	Resume at the earliest unmet dependency.
`STOP`	Boundary verified; return control.	All requested layers and a valid overlay exist.

Both share the GeoState transition boundary absent $\rightarrow$ bound, supported by the returned observation and saved table. Decision-local credit attaches to the establishing action; matched utility compares Skill and OFF at the same eligible state. The State skill returns after its local consequence, whereas the Task skill spans the requested deliverable.

D.2 TEACHER AND COMPILATION

The `o3-mini` teacher reads an evidence packet: a context signature, verified positive executions, policy-failure contrasts, valid negatives, support counts, existing versions and utilities, and the legal tool and role contracts. It returns a structured decision with scope, declared consequence level, trigger, action, argument roles, resources, verification, continuation, repair, stop, anti-trigger, evidence tier, and parent version. Output is at most 2,048 tokens, with one primary request and at most one schema-repair request. The compiler checks public-tool schemas, argument roles, resource dependencies, verification, handoff, repair, stop permission, formula roles, and carrier length. It rejects unresolved fields and unknown parent revisions, then renders the Task or State carrier.

Across 253 evidence packets, the teacher made 267 requests, including 14 schema repairs. It produced 243 candidate SkillSpecs, which the compiler checks before independent admission and paired evaluation.

The declared consequence level L describes the targeted outcome, whereas evidence tier E describes the strongest supporting observation. The State specification above declares L1 with E3 support and renders to a 57-token Qwen3.5 carrier; the Task specification declares L3 with E3 support.

D.3 VERIFIED ADMISSION AND DELIVERY

Compilation establishes structural executability; admission tests realized consequences, and paired utility measures the incremental effect of delivery. Compiler-valid candidates receive preallocated probes at applicable states.

Execution-Verified Admission requires a usable action/resource hand-off (L1), the declared semantic, entity, numerical, unit, or formula relation with provenance (L2), or the specified endpoint/artifact with valid lineage and content (L3). One independent continuation must realize the declared consequence at an applicable instance and pass hard-contract checks. Admission establishes realization, not positive utility.

An admitted NEW version starts with mean utility 0, risk 0, and effective evaluation support 0 for exploration scheduling (Sec. A.9). These neutral values add no observations; posterior ranking begins with paired support. Formal utility pairs may reuse environment initialization, resource snapshots, and prefixes, but count no admission outcome as utility evidence. Eligibility is fixed before treatment, retaining valid failures and successes.

Delivery checks family, frontier, task target, public tools, resource roles, and readiness. Task skills enter at initialization and State skills at applicable frontiers. OFF allocation and ON-version ranking are separate; a negative utility estimate is not itself an OFF-allocation rule, although activation, retention, and safety gates can remove a version from the eligible delivery pool.

D.4 MATCHED TREATMENT EFFECTS

Training records X_g^{train} supply policy gradients, construction evidence, and saved snapshots. At each scheduled utility window, the post-group anchor $\bar{\theta}_w$, candidate set, and eligible snapshots are fixed by the schedule in Sec. A.9. Each evaluated version receives 16 no-update pairs from eight UIDs, with at most two pairs per UID and no more than 16 pairs per version in a window. Subsequent windows prioritize previously unused applicable UIDs for that version; repeated UIDs count once toward cumulative unique-UID support. Utility probes produce no policy gradients; their audited evidence updates the selection posterior.

The estimand is the incremental effect of supplying version v at eligible states under checkpoint $\bar{\theta}_w$. ON and OFF restore separate copies of the same snapshot ζ_t: history, GeoState, resource contents and versions, and tool/environment state. ON supplies v; OFF omits it. Writable packages, layers, and output files are branch-local; continuation seed, verifier, and other-guidance rules are shared. Task skills branch at ζ_0 and State skills at the declared applicable frontier.

Let $\mathcal{P}_{w,v,\ell}$ contain the eligible pairs for one policy window, candidate, and consequence level. It can pool different task UIDs: their saved κ keys need not match across tasks because each pair compares ON and OFF from the same pre-treatment snapshot.

For pair p, define

$$d_p^u = u_{p,\ell}^{\mathrm{ON}} - u_{p,\ell}^{\mathrm{OFF}}, \qquad d_p^b = b_p^{\mathrm{ON}} - b_p^{\mathrm{OFF}}. \tag{26}$$

For arm $a \in \{\mathrm{ON}, \mathrm{OFF}\}$, $b_p^a \in \{0, 1\}$ records whether a prespecified post-branch execution-risk event occurs during that continuation. The difference $b_p^{\mathrm{ON}} - b_p^{\mathrm{OFF}}$ measures incremental risk associated with delivery. Soft-risk events are an invalid or refused action, premature Terminate, repetition with no verified delta, a post-branch tool failure, stagnation, and max-turn termination. Multiple events still count as one. Hard risks (an accepted false STOP, private or gold leakage, forbidden network or path use, destructive persistent-resource mutation, and branch contamination) disable delivery regardless of utility. If the environment is unavailable, $\omega = 0$: the attempt is excluded from utility, support, and the posterior, and retained in the quarantine account. This risk is separate from opportunity reward $r_{i,t}^{\mathrm{opp}}$. The coefficient is $\lambda_{\mathrm{risk}} = 0.5$. Reporting uses the raw delivery effect $\widehat{\Delta U}$; the controller uses risk-adjusted utility only for selection. The two estimates use equal pair weights, with $n = |\mathcal{P}_{w,v,\ell}| > 0$:

$$\widehat{\Delta U}_{w,v,\ell} = \frac{1}{n} \sum_p d_p^u, \qquad \widehat{U}^{\mathrm{sel}}_{w,v,\ell} = \frac{1}{n} \sum_p \left(d_p^u - \lambda_{\mathrm{risk}} d_p^b \right). \tag{27}$$

Each $u_{p,\ell} \in \{0, 1\}$ records verified realization at the declared L1/L2/L3 level. Table 16 illustrates how branch-local risk can change the selection value without changing the reported raw effect.

A candidate-specific ON branch is run for each candidate. One no-guidance OFF continuation is reused only when the policy, snapshot, visible prefix, resources and tool state, seed, other guidance, and execution limits are identical. Each candidate still receives its own OFF outcome $u_{\mathrm{OFF},v}$, scored against that version's declared consequence. A shared OFF execution counts once in the resource ledger, even when it supplies outcomes for several candidate comparisons. Physical execution counts are in Sec. E.11. When utilities from the same UID are compared jointly, one UID-cluster draw is shared across those comparisons; reused OFF outcomes are not treated as independent samples (Fig. 6).

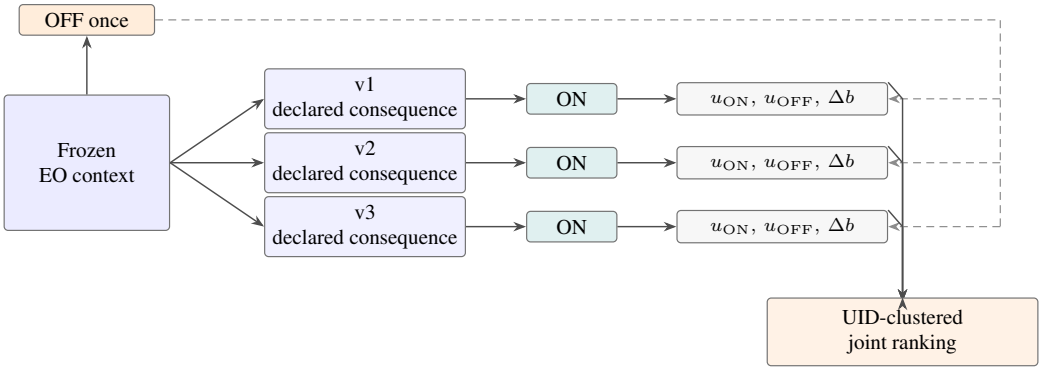

Figure 6: **Shared control execution across candidate comparisons.** Solid arrows are execution or data flow. Dashed arrows reuse the single OFF readout. Three candidates are schematic. Counts are in Sec. E.11.

Table 16: Worked branch-local risk examples from five audited pairs. $U^{\mathrm{sel}} = d^u - 0.5\, d^b$ is used for selection; reported delivery effects use d^u across all enrolled pairs.

Pair	u_{ON}	u_{OFF}	b_{ON}	b_{OFF}	U^{sel}
GIS stale-layer recovery	1	0	0	1	$+1.5$
Spectral preview completion	1	0	0	0	$+1.0$
BBox with extra ON failure	1	1	1	0	-0.5
Repeated no-delta change	0	0	1	0	-0.5
Neutral counting pair	1	1	0	0	0.0

D.5 Evidence-based selection and posterior updates

For one frozen policy window, version, and outcome level, group paired records by task UID (unique task identifier), not by snapshot, rollout, or skill version. Let m_j count pairs in UID cluster j, and define

$$D_j = \sum_{p \in j} d_p^u, \qquad Q_j = \sum_{p \in j} d_p^b. \tag{28}$$

A UID-cluster Bayesian bootstrap, based on the Dirichlet reweighting of Rubin (1981), generates 10,000 utility samples for selection:

$$\boldsymbol{w} \sim \mathrm{Dirichlet}(\mathbf{1}_M), \qquad \widetilde{U}_{w,v,\ell} = \frac{\sum_{j=1}^{M} w_j (D_j - \lambda_{\mathrm{risk}} Q_j)}{\sum_{j=1}^{M} w_j m_j}. \tag{29}$$

Records within a UID share one weight across utility and risk, including candidate comparisons that reuse the same OFF execution. The denominator preserves the pair-weighted estimand, reducing to the risk-adjusted pair mean under uniform weights.

Thompson sampling ranks supported feasible versions within the applicable outcome level using the 10,000 UID-cluster Bayesian-bootstrap samples. NEW uses neutral initialization and least-supported-tier scheduling (Secs. A.9 and D.3). The 4/3/1 rule specifies the nominal exploration allocation; Table 17 sets promotion and retention support.

Each utility distribution is tied to a frozen macro-boundary anchor checkpoint and its evaluated pairs. After a boundary update, the bank and posterior remain fixed throughout the following training interval while the live policy keeps learning. CURRENT uses this anchor-policy posterior, not a utility estimate for every intermediate policy. At the next boundary, fresh paired evidence under the new anchor replaces the previous posterior; older records remain support and provenance history only. Each skill revision is evaluated and tracked separately. The 10,000 selection samples use a Bayesian bootstrap, whereas reported confidence intervals use 10,000 task-UID bootstrap replicates (Sec. E.7).

Support gates and hysteresis. Table 17 follows admitted NEW versions through support accumulation, activation, retention, and selection-safe validation.

Table 17: Version-support gates and delivery eligibility. Pair, unique-UID, and window counts accumulate across windows; utility probabilities use the latest anchor window only. Safety and applicability are required for every delivery. A CURRENT vacancy never creates an extra NEW slot.

Version state	Requirement	Eligible slot
Experimental	Compilation and one independent verified admission	NEW only
TrainBank	≥ 16 pairs, ≥ 8 UIDs, no hard risk	NEW only
Active	≥ 32 pairs, ≥ 16 UIDs, ≥ 2 windows, $P(U_v > 0) \geq 0.90$	CURRENT only
Retained Active	$P(U_v > 0) \geq 0.80$	CURRENT only
Selection-safe	≥ 64 pairs, ≥ 32 UIDs, ≥ 4 windows; 32 validation pairs from 16 RL-DEV-BANK UIDs, excluded from training support and the training posterior	CURRENT only
HOLD / disabled	No policy-training delivery while in either state	Neither

The 0.90 promotion and 0.80 retention probabilities are evaluated using the latest anchor window and provide membership hysteresis. A Selection-safe version also passes an independent check on 32 pairs from 16 RL-DEV-BANK UIDs, two pairs per UID. Those pairs do not enter training support or the training posterior; the validation posterior is fit on them alone. The pass requires $P_{\mathrm{DEV}}(U^{\mathrm{sel}} > 0) \geq 0.90$, a non-negative raw declared-consequence change, L3 endpoint ON at least OFF when that level applies, policy-invalid events on ON no higher than on OFF, at most one extra post-branch tool failure per 32 pairs, and no hard-risk event. Pair and unique-UID support accumulate across windows, with distinct UIDs counted only once. Cumulative support does not enlarge the latest-window utility posterior. Sampling draws $\widetilde{U}_v$ from each CURRENT-eligible applicable version's anchor-specific distribution at the same consequence level and selects $v^\star = \arg\max_{v \in \mathcal{V}_{\mathrm{feasible}}} \widetilde{U}_v$, where $\mathcal{V}_{\mathrm{feasible}}$ contains only these versions. Experimental or TrainBank status does not authorize CURRENT delivery; insufficient support leaves a safely admitted version in NEW exploration, whereas HOLD and disabled versions are excluded from both slots.

Training-utility fitting and Teacher construction use RL-TRAIN records. The Selection-safe validation distribution is estimated on RL-DEV-BANK alone. HELD-OUT, RESERVE (including the 150 composition tasks), and OEA-TEST remain outside promotion, posterior updates, and checkpoint selection (Sec. A.7).

D.6 BANK MEMBERSHIP

Admission places a candidate in the Experimental pool. Experimental and TrainBank candidates may receive the reserved NEW treatment. The active bank B_g contains Active, Retained Active, and Selection-safe procedures eligible for CURRENT, subject to applicability and safety; HOLD, disabled, and superseded candidates are not delivered. Membership and its selection posterior change only at scheduled boundaries, and training uses the most recently committed bank.

The final bank has 47 CURRENT-selectable versions: 8 Selection-safe, 25 Active, and 14 Retained Active. Of these, 24 are Task and 23 are State skills; the State skills comprise 16 recovery and 7

closure procedures. The final L1/L2/L3 composition is 14/18/15. The gates withheld 23 candidates for safety and 46 for insufficient paired support; neither group entered CURRENT.

D.7 Composition across families and evidence levels

Figure 7 organizes the 47 selectable versions by task family, delivery scope, and declared consequence. Each of the seven families contains 4–11 versions and both Task and State guidance; the RL-source task distribution is given in Fig. 5.

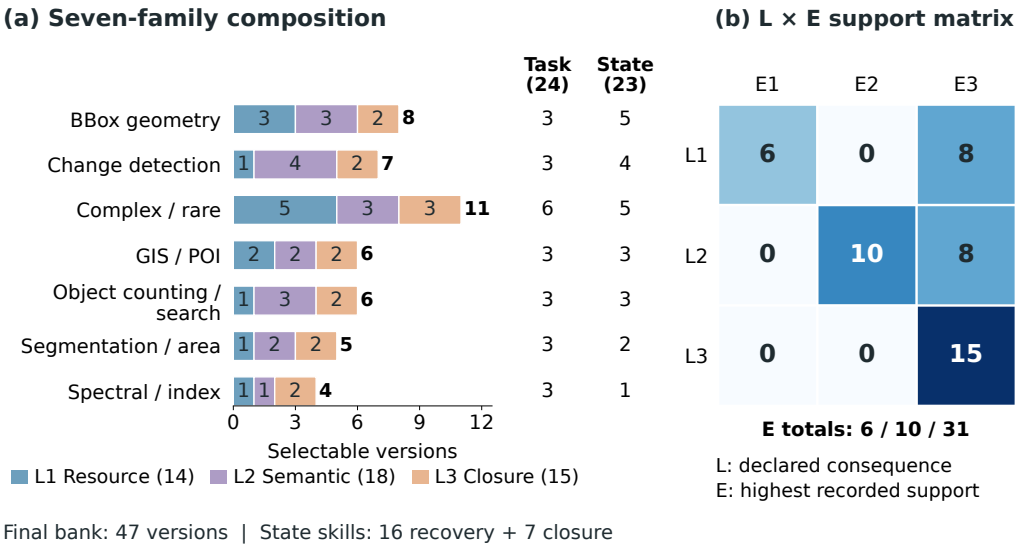

Figure 7: **Composition and recorded support of GeoSkillBank.** (a) Seven-family version counts, partitioned by consequence level, with Task/State delivery counts alongside. (b) Declared consequence level L crossed with the highest evidence tier E recorded per version: resource/predicate (E1), semantic consequence (E2), and endpoint/artifact (E3). Cell counts sum to 47 versions.

E Extended Experimental Analyses

E.1 Native Capability: Evaluation and Scoring

OEA-TEST evaluates end-to-end capability on 1,169 tasks after policy and bank selection are frozen. Actions are executed and later decisions condition on returned observations rather than replaying the reference action sequence. We use the released OpenEarthAgent evaluator without local modifications. The answer judge is `gpt-4o-mini` at temperature 0, with at most 150 output tokens. The returned object has a `Score` in $[0, 1]$ and a `Justification` string. The parser reads the first numeric value before Justification and clamps it to $[0, 1]$; a parse failure scores 0. An API failure retries the same request twice, and an exhausted request leaves that item invalid until rerun. Ans. is 100 times the mean eligible task score. Gen. is 100 times the number of successful matched generation executions divided by 145. Tool selection and ordering use the same released aggregation. The OpenEarthAgent-4B and Llama-3.1-Instruct (8B) rows in main Table 1 reproduce the published OpenEarthAgent Table 4 scores. All other rows are local evaluations; Qwen3.5-9B denotes the unadapted backbone. Local results average three fixed decoding seeds; trained-method comparisons use the prespecified primary training run. Table 34 separately reports independent training repeats.

Parameter sizes in Table 1 use public total counts: DeepSeek-V4-Pro (1.6T), GLM-5.2 (744B), and Kimi-K2.6 (1T). These releases distinguish total sizes from activated parameters: 49B, 40B, and 32B, respectively. GPT-5.5's parameter count is not specified in its official model documentation.

The native metrics capture complementary aspects of capability:

- **Ans./Gen.** Ans. averages graded answer scores on 1,024 eligible non-generation tasks. Gen. is binary success of the matched final generation tool on 145 generation tasks. The reported Gen. score is $100K/(145 \times 3) = 100K/435$, taken over all 435 UID-seed observations.
- **Tool selection.** F1 pools matched, predicted, and reference tool counts within Perception, Operation, Logic, and GIS before computing precision and recall.
- **Tool ordering.** AnyOr. matches the tool multiset regardless of order, SameO. requires the exact sequence including repeated calls, and Uni. compares the unique-tool set. All nine metrics use a percentage scale.

These native scores differ from binary HELD-OUT Answer, task-macro VSC, task-level GEC, one-decision VPR, and blocked-state FRR. Successful generation tool execution verifies execution; physical EO estimates and artifact fidelity require separate validation. Standard GRPO scores 42.50 Ans./64.83 Gen., giving the introductory +7.87/+12.18-point comparison; GRPO+GeoVerify in main Table 2 is the same-feedback condition. Table 9 records the baseline mechanisms and EO substitutions. Their shared 4,328 policy-rollout allowance sets the policy-training budget; total compute and realized teacher/probe usage can differ.

Uncertainty in the nearest baseline comparisons. Table 18 reports paired task-bootstrap intervals for primary-run OEA-TEST score differences. The GDL–SALT intervals include zero, while the TerraWeave–ReSkill intervals are positive on both metrics. These intervals describe task variation conditional on the selected primary checkpoints; independent training variation is reported in Sec. E.11.

For each metric m, the official task type fixes the eligible UID set E_m before method outcomes are observed. Ans. uses graded evaluator scores for non-generation tasks; Gen. uses the evaluator's success indicator for the matched final generation tool on generation tasks. Eligible tasks remain in the denominator after an incorrect answer, absent tool call, malformed output, or early stop, receiving the official evaluator score. The 1,169-task pool is therefore distinct from either metric's eligible denominator. Let $x^A_{i,s,m}$ and $x^B_{i,s,m}$ be the two methods' unit-scale scores for UID i and decoding seed s. The reported difference is

$$\widehat{\Delta}_m = \frac{100}{3} \sum_{s\in\{11,22,33\}} \frac{1}{|E_m|} \sum_{i\in E_m} \left(x^A_{i,s,m} - x^B_{i,s,m}\right). \tag{30}$$

Each of 10,000 bootstrap replicates samples $|E_m|$ UIDs with replacement from E_m and recomputes this difference. Both methods' records for all three decoding seeds accompany each sampled UID; decoding seeds are not resampled separately. A fixed, recorded RNG seed controls the draws. The 2.5th and 97.5th percentiles give the 95% interval. Ans. and Gen. are resampled separately using their respective eligibility sets.

Table 18: **Nearest-baseline differences on OEA-TEST.** Differences and paired task-bootstrap 95% confidence intervals are in percentage points, conditional on the primary training runs.

Comparison	ΔAns.	95% CI	ΔGen.	95% CI
GDL − SALT	+1.13	$[-0.32, 2.55]$	+1.15	$[-0.44, 2.72]$
TerraWeave − ReSkill	+2.26	$[0.49, 4.07]$	+2.76	$[0.83, 4.64]$

E.2 Mechanism Ablations and Grounded Execution

The ablations test which components of the shared execution reference and skill lifecycle support grounded completion. Table 19 specifies the operational replacement in each ablation, while main Table 3 reports the corresponding absolute outcomes. All conditions use frozen GeoVerify on the same 297 HELD-OUT UIDs: Answer/GEC are binary task outcomes, and VSC averages each task's required non-endpoint completion ratio. FRR uses the common blocked-state panel below. These tables give point estimates; repeated-state uncertainty groups all opportunities from each task UID together.

Common blocked-state full continuations. $\mathcal{B}_{\text{full}}$ contains 591 policy-independent blocked-frontier opportunities selected before method evaluation. Each condition restores the same history, GeoState, resources, versions, and tool state, then continues to endpoint, execution limit, or unrecoverable termination. The outcome is subsequent observation-verified restoration of the frozen unmet or violated predicate. Policy and whole-bank setting vary; initial opportunity and verifier are shared.

Every selected state is evaluated even if a policy would not visit it naturally. Invalid or premature actions count as non-restoration unless a later action repairs the predicate; external environment failures censor the state symmetrically. Opportunities receive equal weight, with uncertainty clustered by UID. At θ_C, enabling the whole bank raises restored frontiers from 127/591 to 174/591. This measures recoverability from common blocked states. The separate 591 State-Recovery pairs from 207 UIDs in Sec. E.8 instead hold the policy and Task guidance fixed while varying recovery guidance.

Table 19: **Operational definitions of the mechanism ablations.** Each row states the removed mechanism, its replacement, and the retained components. All conditions use the same frozen GeoVerify evaluator.

Variant	Removed	Replacement	Retained
w/o GeoState	Typed state, verified progress, κ, and state-based routing	Raw history, trajectory-level GRPO, and family/action-pattern routing	Bank evolution, Teacher, safety checks, and budgets
w/o dependency-opportunity matching	Semantic opportunity groups	Task-and-decision-index (q, t) groups	Local rewards, eligibility, action-token loss, and skill evaluation
First-version Bank	Revision and version evolution within existing identities	Permanently retain each identity's first admitted version	New identities, utility estimation, promotion, and policy updates
w/o matched utility	Matched-effect posterior and utility-guided selection	Uniform identity selection; newest version within each identity; support-only promotion	Teacher, compiler, verified admission, routing, and NEW exploration
w/o compiler / verified admission	Typed semantic compilation and execution-based consequence admission	Minimal sanitizer, followed by direct entry into NEW	Runtime safety, routing, matched utility, Thompson selection, and policy updates

GeoState removal changes the learning interface as a bundle; the common GeoVerify evaluator measures its outcomes without supplying state-based credit or routing. Table 19 distinguishes this bundled change from the more targeted replacements.

Dependency-opportunity matching ablation. Within the full co-evolution training setup, this ablation replaces semantic opportunity grouping with task-and-decision-index grouping, (q, t), for policy advantages:

$$A^{\text{abl}}_{i,t} = \frac{r^{\text{opp}}_{i,t} - \mu_{q_i,t}}{\sigma_{q_i,t} + \epsilon}.$$

The statistics pool existing loss-eligible decisions from the same task and decision index, without padding terminated trajectories. They use the same r^{opp} before post-rollout accounting (Sec. C.1); eligibility and action-token optimization are unchanged. Skill–OFF evaluation retains the same saved snapshot, pre-action κ, GeoVerify, resource gates, and STOP conditions. The measured difference is the system-level consequence of replacing the comparison key during co-evolution, including any resulting changes in policy trajectories and bank development. It evaluates dependency-opportunity matching within the full system; policy-only comparisons additionally involve the reward and eligibility rules.

Among the tested replacements, removing the GeoState-based interface yields the largest degradation in grounded execution. The skill-side ablations in Table 20 further separate the roles of applicability routing, execution-verified admission, compilation, and delivery scope.

Table 20: **Additional skill-side ablations on 297 HELD-OUT tasks.** The rows separate applicability, admission, compilation, and delivery scope. Answer reports correct tasks out of 297 with percentages rounded to one decimal; VSC reports verified dependency completion (%).

Variant	Answer ↑ correct/297 (%)	VSC ↑
Full TerraWeave	**150/297 (50.5)**	**69.7**
Applicability and candidate validation		
w/o state-aware applicability routing	143/297 (48.1)	61.8
w/o execution-verified admission	142/297 (47.8)	62.4
w/o executable compiler only	144/297 (48.5)	65.1
Delivery scope		
Task GeoSkills only	146/297 (49.2)	65.8
State GeoSkills only	144/297 (48.5)	63.2

The routing ablation uses task/family-compatible retrieval without frontier, resource-role, or readiness gating. Admission removal sends compiler-valid candidates directly to exploration; compiler-only removal retains the separate execution-verified admission stage. Both delivery scopes contribute, with the larger VSC loss when Task guidance is removed.

E.3 Credit and Completion across Families and Dependency Depth

Distribution of grounded completion. On the 297-task HELD-OUT panel, grounded-completion gains extend across all six routine EO families, while all methods remain at zero on the 11-task Complex / rare subset (Table 21). The similar micro and family-macro improvements indicate that the aggregate gain is not driven only by the largest GIS / POI family. The same-feedback GRPO+GeoVerify condition achieves 35/297 GEC successes, distinct from Standard GRPO's 28/297 in the main mechanism table.

Table 21: **Family-wise grounded endpoint completion.** Cells report successes/tasks (%); differences use the unrounded rates. Micro pools all tasks, whereas family macro weights each of the seven families equally, including Complex / rare.

Family	n	GRPO+GeoVerify	GDL-GRPO	TerraWeave	Δ vs. GDL
GIS / POI	126	19/126 (15.1)	22/126 (17.5)	35/126 (27.8)	+10.3 pp
BBox geometry	78	7/78 (9.0)	8/78 (10.3)	16/78 (20.5)	+10.3 pp
Change detection	27	3/27 (11.1)	4/27 (14.8)	8/27 (29.6)	+14.8 pp
Object counting / search	23	2/23 (8.7)	3/23 (13.0)	5/23 (21.7)	+8.7 pp
Segmentation / area	18	1/18 (5.6)	2/18 (11.1)	5/18 (27.8)	+16.7 pp
Spectral / index	14	3/14 (21.4)	3/14 (21.4)	4/14 (28.6)	+7.1 pp
Complex / rare	11	0/11 (0.0)	0/11 (0.0)	0/11 (0.0)	0.0 pp
Micro	**297**	**35/297 (11.8)**	**42/297 (14.1)**	**73/297 (24.6)**	**+10.4 pp**
Family macro	7	**10.1%**	**12.6%**	**22.3%**	**+9.7 pp**

Common-snapshot VPR protocol. The common-snapshot comparison tests one-decision progress under shared execution opportunities. Before the GRPO/GDL training fork, cold-start θ_0 screens all 297 HELD-OUT UIDs with four fixed collector seeds per UID and SkillBank OFF. A rollout can supply multiple states, saved before either evaluated policy acts. Deduplication uses UID, collector seed, snapshot identity, saved κ, and resource-version signature; equal κ does not merge distinct executable states.

Frozen GRPO+GeoVerify and GDL each receive a separate copy of every eligible pre-action state, with the same evaluation seed, runtime, and verifier and SkillBank OFF. Eligibility is set before treatment. The unit is one decision and its observation: any net-new verified requirement counts as progress, policy-invalid actions score zero, and external environment failures censor both arms symmetrically.

Snapshots are equally weighted and uncertainty clusters all records from a UID. Table 22 gives point estimates for 12,667 eligible snapshots out of 16,240 collected. VPR here measures progress from common states, with the same exposure distribution for both policies.

Progress and completion across dependency depth. GDL-GRPO's VPR advantage grows from 2.6 to 11.6 pp as remaining depth increases, while eligibility coverage falls from 86.0% to 64.0% (Table 22). The separate task-level analysis groups all 297 tasks by maximum depth and also finds its largest GEC gain on depth-≥ 5 tasks (4.0 pp; Table 23). These complementary trends support the value of dependency-conditioned credit in harder workflows; they condition on the observed states and do not identify a causal effect of depth itself.

Table 22: **Decision-level verified progress by remaining dependency depth.** VPR cells report progress outcomes/eligible common snapshots (%), one decision per policy per snapshot. Coverage uses all collected snapshots. Point estimates use equal snapshot weights; uncertainty clusters by UID.

Remaining depth	Eligible / all	Coverage	GRPO+GeoVerify	GDL-GRPO	Δ (pp)
Shallow 1–2	5,767 / 6,706	86.0%	3,010/5,767 (52.2)	3,160/5,767 (54.8)	+2.6
Medium 3–4	4,446 / 5,700	78.0%	1,903/4,446 (42.8)	2,192/4,446 (49.3)	+6.5
Deep ≥ 5	2,454 / 3,834	64.0%	746/2,454 (30.4)	1,031/2,454 (42.0)	+11.6
Overall	12,667 / 16,240	78.0%	5,659/12,667 (44.7)	6,383/12,667 (50.4)	**+5.7**

Table 23: **Task-level grounded completion by maximum dependency depth.** Cells report completed tasks/all tasks in the bin (%). These task bins are distinct from the decision bins in Table 22.

Maximum task depth	Tasks	GRPO+GeoVerify	GDL-GRPO	Δ (pp)
Shallow 1–2	112	20/112 (17.9)	22/112 (19.6)	+1.8
Medium 3–4	110	11/110 (10.0)	13/110 (11.8)	+1.8
Deep ≥ 5	75	4/75 (5.3)	7/75 (9.3)	+4.0
Overall	**297**	**35/297 (11.8)**	**42/297 (14.1)**	**+2.4**

E.4 HELD-OUT Answer Accuracy and Verified Dependencies

Joint outcomes locate answer gains relative to verified prerequisite completion. Across the same 297 HELD-OUT tasks, correct answers with full pre-answer dependency closure increase from 71 under Standard GRPO to 98 under GDL-GRPO, 120 under the co-evolved policy with the bank disabled, and 134 under full TerraWeave. Meanwhile, correct answers without verified closure decrease from 52 to 16 (Table 24). Thus, improvements in answer accuracy increasingly coincide with observation-supported dependency closure.

Table 24: **Joint answer correctness and dependency closure on 297 HELD-OUT tasks.** All entries are percentages over the same task panel.

Condition	Answer	Dependency closure	Correct + closure	Correct without closure
Standard GRPO	41.4	34.0	23.9	17.5
GDL-GRPO, OFF	45.5	44.1	33.0	12.5
Co-evolved, OFF	48.8	52.9	40.4	8.4
TerraWeave	**50.5**	**59.9**	**45.1**	**5.4**

E.5 Crossed Policy–Bank Evaluation

The same evolved $B^\star$ is evaluated with θ_P and θ_C on OEA-TEST under paired task and decoding conditions. Each selected policy is frozen and compared with bank OFF and ON; the native metrics retain the eligibility and aggregation in Sec. E.1. For metric m, let S_m denote the score and define

$$\Delta_{B,m}(\theta) = S_m(\theta, B^\star) - S_m(\theta, \mathrm{OFF}). \tag{31}$$

Bank ON raises native Ans./Gen. at both frozen checkpoints (Tables 25 and 26). With the bank OFF, θ_C exceeds θ_P by 3.15/6.44 points, retaining 72.7%/84.8% of the post-GDL Ans./Gen. improvement.

Table 25: Crossed OEA-TEST scores (%). ± denotes standard deviation across three decoding seeds at each selected policy–bank condition. $\theta_P + B^\star$ is an evaluation condition.

Condition	Ans.	Gen.	Per. F1	Op. F1	Logic F1	GIS F1
θ_P, OFF	46.04±0.94	69.43±1.59	38.88±1.44	51.76±0.96	39.48±1.67	88.03±0.59
$\theta_P + B^\star$	46.84±0.86	70.34±1.38	40.02±1.42	52.71±0.94	40.18±1.60	88.67±0.57
θ_C, OFF	49.19±0.80	75.86±1.38	39.34±1.37	57.75±0.89	40.55±1.56	92.10±0.53
$\theta_C + B^\star$	50.37±0.78	77.01±0.80	53.05±1.32	60.68±0.85	45.16±1.47	96.98±0.48

Table 26: Bank gains on the primary OEA-TEST metrics (pp). Ans. brackets are paired 95% intervals; Gen. gains are point estimates.

Metric	Bank gain at θ_P	Bank gain at θ_C
Ans.	+0.80 [0.12, 1.48]	+1.18 [0.41, 1.95]
Gen.	+0.92	+1.15

Decoding SDs summarize three-seed dispersion at each selected checkpoint; paired UID-bootstrap intervals describe task variation in Ans. bank effects conditional on these policies and bank. Independent training variation is evaluated separately in Sec. E.11.

E.6 Frozen-Bank Transfer and 4B Policy Evaluation

Frozen-bank transfer. The final $B^\star$ is transferred to nine frozen target policies on OEA-TEST. Each target is compared with its own locally evaluated Bank-OFF condition under common task/decoding settings, without parameter updates. ON delivers the applicable compact Task/State carrier into the target policy context through the common delivery interface; OFF omits it. These transfer results measure the effect of deploying the supplied procedure, including its additional context; they do not isolate a token-length-controlled semantic effect. For every target, Bank OFF and ON use the same 145 generation-task UIDs and seeds 11, 22, and 33. K^{OFF} and K^{ON} count successful final-generation executions among the 435 task–seed records. Each arm is $100K^a/435$, and the bank effect is $100(K^{\mathrm{ON}} - K^{\mathrm{OFF}})/435$. Missing, malformed, early-stopped, or incorrect executions stay in the denominator with zero success. Ans. remains the graded judge score.

Reuse is receiver-dependent: Qwen3-32B regresses on Logic/SameO./Gen., OpenEarthAgent-4B on ordering, and unadapted Qwen3.5-9B on Ans./Gen. The success counts behind the Gen. column are DeepSeek-V4-Pro 324/330, GPT-5.5 291/302, GLM-5.2 333/336, Kimi-K2.6 297/299, Qwen2.5-7B 201/205, Qwen3-32B 99/97, OpenEarthAgent-4B 330/335, Qwen3.5-9B 170/169, and Qwen3-4B 51/51 ($K^{\mathrm{OFF}}/K^{\mathrm{ON}}$ out of 435).

4B policy evaluation. A separate experiment applies cold-start SFT and grounded RL to a 4B backbone. Table 28 uses native OEA-TEST aggregation across tasks and decoding seeds, with the unadapted instruction model as its initial reference and OpenEarthAgent-4B as an external comparison. These point scores evaluate combined alignment and RL rather than frozen-bank transfer; independent-training uncertainty is not estimated.

The 4B TerraWeave policy improves the unadapted instruction model across all six metrics. Against the reported OpenEarthAgent-4B reference, it is higher on Per./Op./Ans. and lower on Logic/GIS/Gen. The mixed comparison suggests that improved answer quality need not coincide with better artifact generation.

Table 27: **Frozen GeoSkillBank transfer across target policies on OEA-TEST (1,169 tasks).** Entries are $\Delta_B = \text{ON} - \text{OFF}$ in percentage points. Gen. differences use unrounded success counts on the same 145×3 records as native evaluation. Ans. is the graded judge score.

Model	ΔPer.	ΔOp.	ΔLogic	ΔGIS	ΔAnyOr.	ΔSameO.	ΔUni.	ΔAns.	ΔGen.
DeepSeek-V4-Pro	+7.99	+17.50	+5.18	+3.93	+2.48	+2.82	+3.17	+4.03	+1.38
GPT-5.5	+5.74	+9.19	+4.35	+1.15	+4.46	+4.15	+4.52	+2.89	+2.53
GLM-5.2	+2.11	+5.72	+1.71	+1.89	+4.39	+4.51	+4.08	+0.35	+0.69
Kimi-K2.6	+1.46	+5.36	+1.43	+1.68	+4.92	+4.76	+4.43	+0.30	+0.46
Qwen2.5-7B	+1.95	+4.58	+1.18	+2.25	+4.84	+4.79	+4.67	+0.38	+0.92
Qwen3-32B	+1.64	+2.62	−0.42	+1.68	+1.15	−0.18	+1.05	+0.74	−0.46
OpenEarthAgent-4B	+1.82	+5.08	+1.88	+0.50	−0.27	−0.03	−0.35	+0.92	+1.15
Qwen3.5-9B	+1.58	+2.54	+0.51	+1.15	+4.24	+4.05	+3.08	−0.15	−0.23
Qwen3-4B	+0.14	0.00	+0.12	−1.44	−1.28	−0.98	−1.28	−0.69	0.00

Table 28: **TerraWeave policy learning with a 4B backbone.** The first row is the unadapted instruction model; the final row includes cold-start alignment and grounded RL. OpenEarthAgent-4B is an external reference. All values are percentages.

Model / condition	Per. F1	Op. F1	Logic F1	GIS F1	Ans.	Gen.
Qwen3-4B-Instruct-2507	13.39	29.03	6.93	59.72	6.66	11.72
OpenEarthAgent-4B	58.30	56.76	51.18	98.52	45.26	75.86
TerraWeave-4B policy	**58.49**	**59.34**	46.45	96.94	**46.86**	71.72

E.7 Controlled Effects on Resources, Semantics, and Closure

The crossed evaluation measures the effect of enabling the bank as a whole at a frozen policy checkpoint. We next isolate individual skill deliveries by restoring the same executable state and varying only the assigned guidance.

Matched interventions test three complementary binary consequences of skill delivery. L1 verifies the declared resource/action transition, L2 the required semantic, entity, value, and unit consequences, and L3 grounded endpoint or artifact closure. Each level has its own applicability set, so the rates measure distinct outcomes rather than a cross-level capability ranking.

Pre-treatment enrollment and version selection. The collector freezes the selected co-evolved policy $\theta_C^\star$ with the bank off, on HELD-OUT, after policy and bank selection and before ON/OFF outcomes are observed. Eligible versions are those that pass family, scope, consequence level, frontier, resource-role, and readiness gates and have no hard risk. Within each identity we keep the latest eligible version, then rank by the frozen training-only posterior mean:

$$v^\star(\zeta_t, \ell) = \underset{v \in \mathcal{V}_{\text{safe}}(\zeta_t, \ell)}{\arg\max} \; \bar{U}_v^{\text{TRAIN}}.$$

Ties break by greater matched-pair support, greater unique-UID support, earlier admission, and a fixed version order. No Thompson draw and no HELD-OUT outcome enters this ranking. The CURRENT-selectable pool is Active, Retained Active, and Selection-safe; the eight Selection-safe versions are the subset that also passed independent validation.

Scope-aware intervention units. Each UID contributes four matched pairs. A Task intervention uses one sealed initialization snapshot with four prespecified continuation seeds, whereas a State intervention uses four distinct eligible frontier snapshots with one continuation seed each (Table 29). Enrollment is sealed before either branch outcome is observed.

Matched continuations and counting. Each pair shares θ_C, task UID, the saved state, and other-guidance rules. ON supplies the assigned version; OFF withholds it. Writable resources are branch-local. The L1/L2/L3 effects therefore describe skill delivery within the enrolled applicable population; they are not averages over all HELD-OUT tasks or every candidate version.

Table 29: Scope-aware matched-pair panels. Each UID contributes four pairs.

Level	UIDs	Pairs	Task UIDs/pairs	State UIDs/pairs	Snapshots	Versions
L1	100	400	40/160	60/240	280	14
L2	150	600	75/300	75/300	375	18
L3	200	800	120/480	80/320	440	15

Let $u_{i,\ell}^{\mathrm{ON}}, u_{i,\ell}^{\mathrm{OFF}} \in \{0,1\}$ and $D_{i,\ell} = u_{i,\ell}^{\mathrm{ON}} - u_{i,\ell}^{\mathrm{OFF}}$. Here $W = N_{10}$ counts ON-only successes and $L = N_{01}$ counts OFF-only successes; N_{11} and N_{00} are ties. Every pair has equal weight:

$$n_\ell = N_{11} + W + L + N_{00}, \qquad \widehat{\Delta U}_\ell = 100\,\overline{D}_\ell = 100\frac{W-L}{n_\ell}. \tag{32}$$

Equal pair weighting also equals the mean of the four-pair UID effects; there is no further seed, UID, or version weighting. Main Sec. 4.4 reports OFF/ON rates, whereas Table 30 reports intervals for their paired difference.

Paired effects and confidence intervals. We resample task UIDs in 10,000 bootstrap replicates, retaining all seed, context, and version records for each sampled UID. Each replicate recomputes the pair-weighted mean; the 2.5th and 97.5th percentiles form the 95% interval. This effect-reporting bootstrap is separate from the risk-adjusted Bayesian bootstrap used for skill selection.

Table 30: Matched effects and task-UID bootstrap intervals, estimated separately for the resource, semantic, and endpoint outcome sets. Each panel uses pre-enrolled pairs and deterministic version assignment (Sec. E.7).

Level / outcome	OFF (%)	ON (%)	ΔU (pp)	95% CI (pp)	n	UIDs
L1 Resource	58.0	67.0	+9.0	$[+5.0, +13.0]$	400	100
L2 Semantic	46.0	53.0	+7.0	$[+3.5, +10.5]$	600	150
L3 Closure	15.5	20.5	+5.0	$[+2.2, +7.8]$	800	200

E.8 Matched Repair-to-Endpoint Outcomes

State Recovery guidance improves restoration and subsequent grounded closure under a frozen policy. We evaluate RL-DEV-selected θ_C on 297 HELD-OUT UIDs; 207 contain a policy-controllable, skill-applicable repair opportunity, yielding 591 matched pairs. Each restores separate copies of the same pre-treatment ζ_t with a shared UID and continuation seed. OFF withholds State Recovery guidance; ON supplies the matching State Recovery Skill from frozen $B^\star$. Task guidance and prior observations remain fixed. This recovery-specific control differs from whole-bank OFF in $\mathcal{B}_{\mathrm{full}}$ and native evaluation.

For a branch of pair p, let f_p indicate restoration of the blocked frontier. Repair-to-Endpoint additionally requires subsequent grounded closure of the requested task:

$$e_p = \mathbf{1}[\text{frontier restored and requested grounded endpoint reached}], \qquad e_p \le f_p.$$

FRR and R2E weight all 591 pairs equally. Confidence intervals resample the 207 task UIDs with all associated pairs. Recovery-specific OFF restores 108 frontiers; the separate whole-bank comparison restores 127 with bank OFF and 174 with full TerraWeave (main Table 3). The common 591-state panel and this 207-UID recovery panel have distinct enrollment and controls.

Guidance helps repair blocked dependencies and carry more executions through to the requested endpoint. The conditional endpoint rate describes what happens after repair; its denominator differs between arms because restoration itself is affected by guidance. The unconditional paired R2E effect measures the resulting improvement in completed workflows.

E.9 Executed-Tool Failure Profile

With the final bank enabled under the frozen co-evolved policy, failed tool calls decrease from 1,297/6,580 to 1,072/6,421, reducing the pooled failure rate from 19.71% to 16.70% (Table 32). The

Table 31: Matched restoration and Repair-to-Endpoint under frozen θ_C. Both unconditional outcomes use 591 pairs from 207 UIDs. Rates are percentages; effects and task-UID bootstrap 95% intervals are percentage points.

Outcome	Recovery OFF	Recovery ON	Δ	95% CI
Frontier restored	108/591 (18.3)	138/591 (23.4)	+5.1	—
R2E	82/591 (13.9)	107/591 (18.1)	+4.2	$[1.7, 6.8]$
Endpoint | restored	82/108 (75.9)	107/138 (77.5)	+1.6	—

remaining failures are highly concentrated in ChangeDetection, which accounts for 690 of the 1,072 failed calls under Bank ON. These receipt-level diagnostics include environment errors and recovered retries, and therefore characterize observed execution failures rather than policy-attributable errors alone.

Table 32: Complete final-bank executed-tool failure profile. Rates use each tool's executed calls as the denominator; differences use unrounded rates. Raw failures include environment errors.

	Failed / executed		Rate (%)		Δ (pp)
Tool	OFF	$+B^\star$	OFF	$+B^\star$	ON$-$OFF
OCR	0/19	0/24	0.00	0.00	$+0.00$
TextToBbox	129/656	92/625	19.66	14.72	-4.94
ImageDescription	6/65	5/67	9.23	7.46	-1.77
RegionAttributeDescription	20/382	16/370	5.24	4.32	-0.91
CountGivenObject	1/106	1/112	0.94	0.89	-0.05
ChangeDetection	755/889	690/866	84.93	79.68	-5.25
SegmentObjectPixels	5/76	4/82	6.58	4.88	-1.70
ObjectDetection	0/2	0/3	0.00	0.00	$+0.00$
DrawBox	2/102	1/100	1.96	1.00	-0.96
AddText	2/40	1/42	5.00	2.38	-2.62
GoogleSearch	0/116	0/121	0.00	0.00	$+0.00$
Calculator	17/662	14/640	2.57	2.19	-0.38
Plot	3/12	3/15	25.00	20.00	-5.00
Solver	9/274	7/268	3.28	2.61	-0.67
GetAreaBoundary	65/770	48/758	8.44	6.33	-2.11
AddPoisLayer	224/1,299	150/1,245	17.24	12.05	-5.20
ComputeDistance	28/534	20/522	5.24	3.83	-1.41
DisplayOnMap	1/94	1/88	1.06	1.14	$+0.07$
AddIndexLayer	28/276	16/260	10.14	6.15	-3.99
ComputeIndexChange	0/124	1/126	0.00	0.79	$+0.79$
ShowIndexLayer	0/29	0/31	0.00	0.00	$+0.00$
GetBboxFromGeotiff	2/33	2/34	6.06	5.88	-0.18
DisplayOnGeotiff	0/20	0/22	0.00	0.00	$+0.00$
Overall	1,297/6,580	1,072/6,421	19.71	16.70	-3.02

E.10 Learning Dynamics on the Fixed RL-DEV Panel

The fixed RL-DEV panel tracks retained policy learning and contemporaneous CURRENT-bank guidance at 11 scheduled checkpoints in the primary run (Fig. 8). Co-evolved OFF and Current Bank ON use the same policy checkpoint, scorer, decoding seeds, and execution conditions. GEC uses all 202 tasks as its denominator; VPR uses eligible decisions encountered during validation. These are checkpoint observations from one training run; independent replications appear in Sec. E.11.

The x-axis follows the scheduled policy-rollout budget (Table 33); Phase II begins at 74.1% of that allowance. Each boundary evaluates the post-update anchor with the bank committed after its utility window. CURRENT guidance first becomes available after the second utility window and only where an applicable Active skill exists; otherwise ON provides no guidance at that decision.

At the final scheduled checkpoint, enabling the bank for the co-evolved policy raises full-panel GEC from 35/202 to 44/202 and VPR from 55.6% to 61.4%. GDL-GRPO OFF is 27/202 and Standard GRPO is 21/202. The OFF/ON gap at the same endpoint shows how runtime guidance complements the improvement retained in the policy, consistent with the crossed evaluation in Sec. 4.4. Checkpoint selection instead uses GEC on the 138-UID RL-DEV-POLICY subpanel (Sec. A.10).

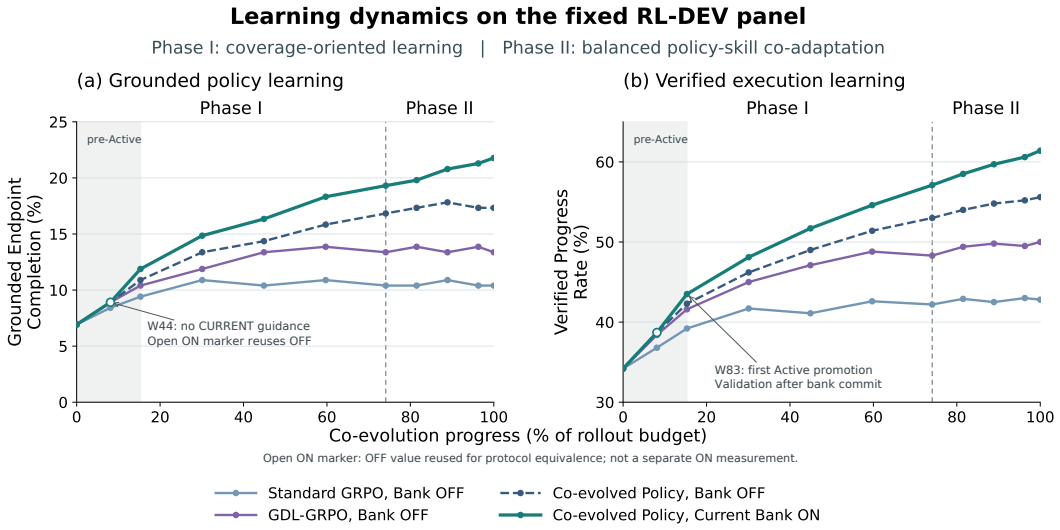

Figure 8: **Learning dynamics on the RL-DEV panel.** (a) Grounded endpoint completion; (b) verified progress rate. The plotted counts use the 202-UID panel. Selection uses the 138-UID policy subpanel. The dashed line is the Phase-I/II boundary at 74.1%. At the first boundary, the open ON marker equals OFF because no CURRENT skill is active; connecting segments guide the eye.

Table 33: Scheduled checkpoints on the policy-rollout budget axis. The rollout allowance is $8g$; optimizer updates can be skipped. The 4,377 bank-evidence executions are accounted for separately. Phase II begins after group 401, at 74.1% of the allowance.

Group g	0	44	83	163	243	323	401	441	481	521	541
Budget (%)	0.0	8.1	15.3	30.1	44.9	59.7	74.1	81.5	88.9	96.3	100.0
Policy rollouts	0	352	664	1,304	1,944	2,584	3,208	3,528	3,848	4,168	4,328

E.11 Training Robustness, Composition, and Execution Cost

Independent training runs. GDL, ReSkill, and TerraWeave each have a primary training run (master seed 0) designated before OEA-TEST evaluation and three independent replications (master seeds 1, 2, and 3). The primary runs supply the main-table scores. Replications retain the data, initialization, training, and selection protocol in Appendix A, changing only the training master seed. GDL and TerraWeave use corresponding seeds; GDL keeps the bank OFF.

Table 34 averages decoding seeds 11, 22, and 33 on the same 1,169 OEA-TEST tasks for each frozen policy. TerraWeave exceeds GDL and ReSkill in Ans. and Gen. for all three reported run indices. SALT and crossed-checkpoint retained-policy fractions use the primary run.

Learning-level compositional holdout. The 150-task holdout drawn from 400-UID RESERVE tests unseen compositions of known tools and requirements. Its task instances, complete source trajectories, designated critical edge pairs, and three-node motifs are excluded from SFT, RL, Teacher evidence, admission, and training-time utility. The exclusion key is the tuple of frozen template identity, canonical node IDs, directed edge set, and role bindings, so an atomic edge, an edge pair, and a three-node motif are distinct. The 150 tasks are GIS/POI 63, BBox 39, Change 14, counting/search

Table 34: Three additional independent training runs per method. Run 1/2/3 uses master seed 1/2/3. Each Ans./Gen. entry averages decoding seeds 11, 22, and 33. The final column reports mean ± sample SD across the three training-run means, excluding primary seed 0.

Method	Run 1	Run 2	Run 3	Mean ± SD
GDL	46.39/69.89	47.08/70.80	47.61/71.49	47.03 ± 0.61 / 70.73 ± 0.81
ReSkill	47.79/73.79	48.63/74.94	48.28/74.48	48.23 ± 0.42 / 74.41 ± 0.58
TerraWeave	50.48/77.24	51.07/78.16	51.43/78.85	50.99 ± 0.48 / 78.08 ± 0.81

12, segmentation/area 9, spectral/index 7, and complex/rare 6. Representative held-out combinations include a boundary, two POI layers, and a distance; two index layers, change detection, and three previews; a localized box, geometric value, and annotated artifact; and a mask, physical area, and unit-normalized endpoint. Frozen TerraWeave answers 74/150 correctly (49.3%) versus 63/150 (42.0%) for ReSkill: +7.3 points with task-UID-bootstrap 95% CI $[2.0, 12.7]$. VSC is 66.47 versus 57.28 (+9.19 points). Tasks receive equal weight and uncertainty resamples UIDs. These results support reuse across unseen combinations of familiar dependencies.

Execution accounting. Policy training uses 4,328 full rollouts; bank evaluation adds 4,377 executions (Table 35). The $1.28\times$ accelerator-use ratio in Table 36 excludes API, CPU, and tool-server time. These measurements distinguish policy-training exposure from total execution cost.

Table 35: Execution activity. Suffix probes do not update the policy.

Activity	Count	Gradient
Policy-training rollouts	4,328	eligible groups
Candidate ON suffixes	3,072	no
Shared OFF suffixes	840	no
Unmatched or censored probes	216	no
Admission probes	243	no
Quarantined attempts	6	no

Table 36: Measured accelerator use for one training seed. API, CPU, and tool-server time are not folded into these GPU-hours.

Run	A800 GPU-hours
GRPO reference	228.0
TerraWeave	291.8
Ratio	$1.28\times$

F EO Execution Cases

These cases illustrate how tool observations support persistent resources and completed outputs; they do not estimate aggregate effects. GIS 1054 and bbox 1142 trace resource handoffs and task completion, while the spectral cases compare preview delivery across indices. Aggregate matched skill-delivery effects are reported in Sections E.7 and E.8.

F.1 Stored procedures across seven families

Figure 9 summarizes representative Task and State procedures through their intended tasks, guidance, and required checks. The source data retain 24 Task carriers and 23 State carriers, 47 selectable versions in total, together with their family mappings. The following cases connect these procedure contracts to recorded executions and saved artifacts.



Figure 9: **Stored skill procedures across seven EO families.** Cards summarize family-level Task and State guidance. Checks specify the consequence required for continuation or closure. Wording is condensed for display; the stored procedures retain the full guidance.

F.2 Persistent resource binding in GIS (1054)

Figure 10 details the resource handoffs behind the unadapted-backbone (Raw) and TerraWeave executions in Fig. 4.

The task overlays hospitals, clinics, and bus stops on a GeoTIFF. GetBboxFromGeotiff returns $(75.8243, 26.9194, 75.8320, 26.9262)$, ordered (W, S, E, N) (west, south, east, north). Raw prefixes these coordinates with `bbox:`; GetAreaBoundary interprets this as a geocoder query and returns zero results. DisplayOnGeotiff then reports a missing GeoPackage, leaving the final package and image fields empty.

TerraWeave supplies the accepted parenthesized tuple. GetAreaBoundary writes `area_boundary`; AddPoisLayer persists `hospitals`, `clinics`, and `bus_stops` with one, two, and three features. DisplayOnGeotiff creates a 1435×1435 overlay from these layers and the source GeoTIFF. Independent inspection confirms the four package tables and POI counts. This trace shows how the persisted boundary package supports both POI construction and subsequent rendering.



Figure 10: **GIS resource binding and overlay delivery.** The execution records trace boundary creation, persisted POI layers, and rendering. The displayed overlay is the output saved by the shown TerraWeave execution. Rows align semantic stages, while Call labels retain recorded invocation indices. Figure 4 condenses these actions into the four-condition comparison; the frozen-policy matched recovery protocol and aggregate outcomes appear in Sec. E.8.

F.3 Spectral preview delivery (0971, 0085, and 1099)

The spectral cases separate a valid index-change computation from delivery of the requested early-date, late-date, and difference previews.

In Landiras NBR (0971; June/October 2022), boundary construction, both monthly layers, and ComputeIndexChange succeed, but DisplayOnGeotiff rejects its input image. In Amboseli NDVI (0085; 2019/2025) and Catania NDBI (1099; 2019/2024), the display tools cannot resolve the spectral layers in their namespace. ShowIndexLayer provides the appropriate rendering path for these computed layers.

For NBR, the state-level preview-completion skill applies after index layers exist, verifies the three outputs, retries a missing preview using the computed indices, and returns control after L3 delivery. The requirement graph records temporal lineage in the base layers, the derived layer in ComputeIndexChange, and one output requirement per preview. Across NDVI/NBR/NDBI, the baseline saves one/zero/one of the three requested previews; TerraWeave saves all three in each case (Figs. 11–12).

Both executions agree on the NBR and NDVI statistics; the observed difference concerns delivery of the requested preview set. This illustrates the artifact-completion requirement in Sec. 3.1.

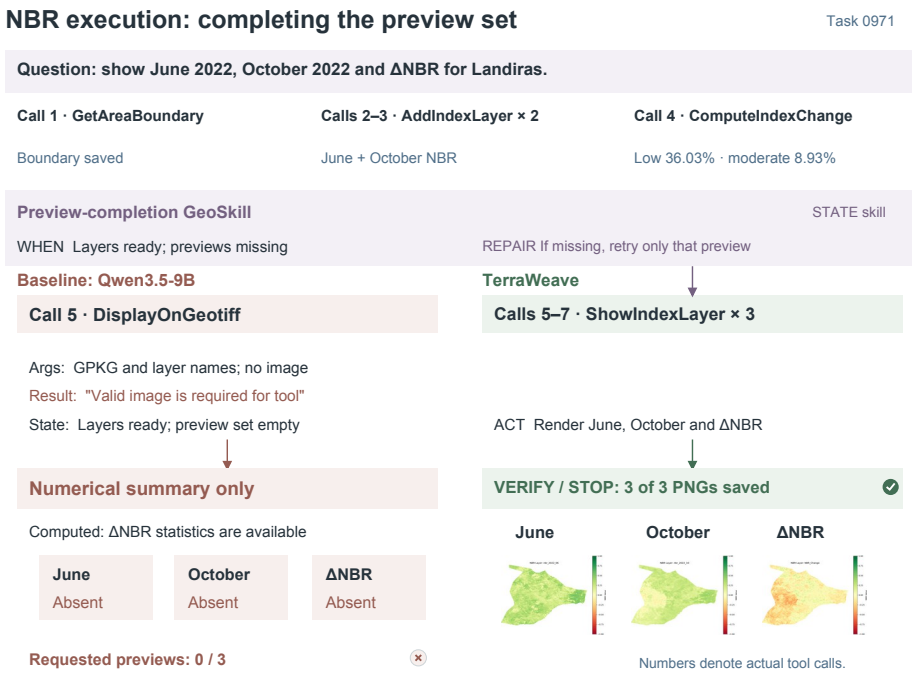

Figure 11: **Completing the requested NBR preview set.** Both runs compute the change statistics. The baseline delivers no previews; TerraWeave saves June, October, and difference PNGs. The state-level preview-completion procedure verifies all three outputs and conditionally retries a missing preview. Call labels retain the recorded tool-invocation indices.

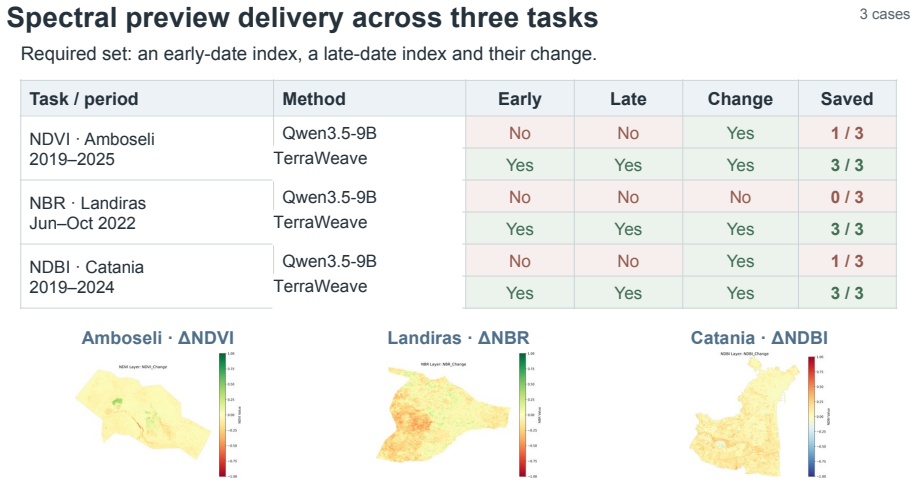

Figure 12: **Role-specific spectral preview delivery.** The matrix identifies which early-date, late-date, and change previews are saved by each method. Totals count these three requested files. The gallery shows TerraWeave change previews with their original scales.

F.4 Recorded bounding-box and artifact verification (1142)

Figure 13 illustrates the bank-OFF behavior examined in Sec. 4.4: both runs record zero skill exposures, selections, and adoptions. Following unsuccessful detections, TerraWeave creates a verified artifact; completion at decision five ends the trajectory without Terminate. Verification

checks nonempty readable output, typed coordinate provenance, source and action bindings, content consistency, and lineage. Independent intersection-over-union accuracy is not measured.

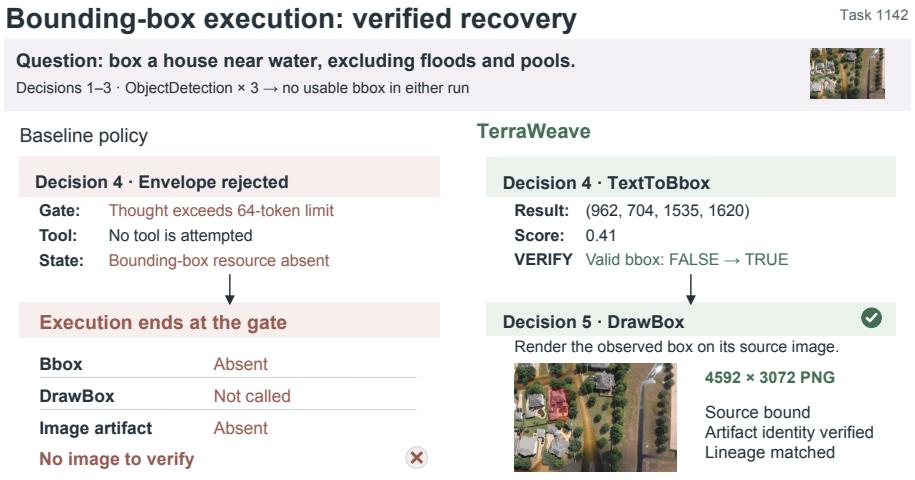

Figure 13: **Verified bounding-box recovery.** Same-seed executions with no delivered skills differ in whether recovery produces a verified artifact. Decision numbers are one-based.