DSA: PROGRESS INSIDE THE GROUP STANDARDIZER IN SPARSE-REWARD AGENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse terminal rewards can silence a group-relative learner even when the states inside a comparison group differ in their proximity to the goal: equal returns are mapped to equal, zero-centered coefficients. We introduce Dense Signal Augmentation (DSA), an advantage operator that combines action-before state progress with return *before* applying the inherited group standardizer. This placement preserves the original grouping and optimizer, recovers the baseline exactly when augmentation is disabled, and exposes progress variation without defining a second normalization scale. A conditional return identity explains the temporal alignment, while a score-function analysis identifies why current-batch scheduling, sampled grouping, and nonstationary proxies prevent an invariance or unbiased-credit guarantee. Predeclared matched training experiments exhibit earlier held-out attainment on average, although the exact paired evidence remains statistically inconclusive. Matched operator controls further test signal placement and temporal alignment, while history-aware and cross-setting experiments delimit deployment and breadth. The principal contribution is therefore a controlled and falsifiable interface between state progress and group-relative credit assignment, supported by training interventions rather than snapshot diagnostics.

1 INTRODUCTION

Group-relative objectives extract empirical update coefficients without a separate value model. For multi-step agents, step-level variants compare actions from repeated anchor observations (Feng et al., 2025; He et al., 2026; Wang et al., 2026). Yet if all trajectories represented in an anchor group have the same sparse terminal return, its normalized step coefficient is zero. A large batch can therefore have little variance-bearing support, though nonzero state weights would not by themselves constitute unbiased action credit.

Zero return variance need not mean equal promise: the same observed anchor can follow histories with different progress, inventory, or goal distance. Signal placement is critical. Separate normalization changes channel scaling; next-state progress shifts coefficient timing; and a new group key changes the comparison itself. We ask whether progress can induce coefficient contrast while preserving the inherited estimator.

DSA changes only the quantity standardized: it combines return with action-before current-state progress, then applies the inherited standardizer once. Separately normalizing the channels is a different estimator; disabling progress recovers the baseline elementwise (Section 3).

The structural failure does not require a dataset-level diagnostic: whenever a realized group has constant return, its return-only standardized coefficient is identically zero. DSA changes that realized coefficient vector when aligned progress varies. Whether this algebraic availability improves the policy is determined only by matched training runs.

This view gives a sharper empirical hypothesis than "more shaping helps": DSA should alter more coefficients before successes create return variance; whether that helps learning is empirical, and any benefit need not persist once returns vary. We therefore prioritize first attainment of a modest held-out threshold, report later and final endpoints without promotion, and treat agreement with this temporal pattern as suggestive rather than causal.

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.

Concurrent estimators also use progress (Fan & Liu, 2026; Yang et al., 2026). Their placement and branching differ, although one fallback can align locally with DSA in a constant-return group. We claim the combined-normalization construction, not generic progress activation or precedence (Section 2).

Contributions.

- **Operator.** We define a combine-before-normalization coefficient operator with exact current-state alignment, fixed groups and optimizer, and an elementwise disabled baseline.
- **Theory boundary.** We derive a conditional return identity and identify why current-batch scheduling, standardization, and learned proxies preclude a DSA invariance theorem.
- **Interventional evidence.** We evaluate four predeclared matched training pairs, then probe the proposed placement with matched operator controls, a history-aware proxy, and separate model-scale and environment tests.

Executable checks establish the operator identity, not its utility. The training experiment shows a directional early-attainment pattern, but limited replication, discrete evaluation cadence, and inconclusive paired tests preclude a confirmed task effect or general superiority.

Relationship to a parallel anonymous submission. A parallel anonymous submission, *Same Groups, Different Values: Decomposing Step-Level Credit Assignment for LLM Agents*, studies partition/value-functional factorization and audits recorded estimator outputs; it neither proposes nor evaluates the DSA training operator. The present paper's task-effect and mechanism claims rely only on its own predeclared matched training experiments, operator controls, proxy tests, and setting-specific replications. Snapshot diagnostics and replay-fidelity results from the Measurement study are not used as evidence for DSA's task effect.

2 Related work

Policy gradients and group-relative credit. DSA inherits the policy-gradient view of return-weighted actions (Williams, 1992; Sutton & Barto, 2018) and advantage-based variance reduction (Schulman et al., 2016). GRPO normalizes rewards within prompt groups (Shao et al., 2024); GiGPO adds anchor-state step groups (Feng et al., 2025), HGPO adds history-conditioned memberships (He et al., 2026), and G2PO constructs a state-transition graph (Wang et al., 2026). Those methods primarily change *who is compared* or the attached value machinery. DSA fixes the inherited group and changes only the scalar entering its standardizer, enabling an elementwise disabled identity and an attributable baseline-versus-DSA test. Auditing whether two estimators share a partition or value functional is a complementary measurement problem; it is outside DSA's algorithmic claim.

Sparse feedback, coverage, and data selection. Exploration methods seek informative experience through counts, prediction error, randomized value functions, episodic memory, or archival return (Bellemare et al., 2016; Pathak et al., 2017; Burda et al., 2019; Osband et al., 2016; Badia et al., 2020; Ecoffet et al., 2021). Replay and relabeling instead reuse or reprioritize experience (Lin, 1992; Mnih et al., 2015; Schaul et al., 2016; Andrychowicz et al., 2017), while offline RL controls extrapolation from a fixed dataset (Levine et al., 2020; Fujimoto et al., 2019; Kumar et al., 2020; Kostrikov et al., 2022). DSA is complementary: it neither acquires nor selects trajectories, but changes normalized coefficients within a fixed batch. Curriculum and level-replay methods instead change which training cases are sampled (Bengio et al., 2009; Jiang et al., 2021); our intervention leaves the task registry and sampling contract fixed while changing the coefficient operator.

Language-agent environments. Text-game agents have long exposed delayed language-action credit (Narasimhan et al., 2015; He et al., 2016; Yuan et al., 2018; Hausknecht et al., 2020; Côté et al., 2018); ReAct interleaves language reasoning and action (Yao et al., 2023). ALFWorld and ScienceWorld provide embodied and scientific testbeds (Shridhar et al., 2021; Wang et al., 2022). We use these environments to test the sparse-return mechanism separately from task gain, not to claim cross-domain generalization.

Potential-based shaping. Fixed-potential shaping preserves optimal policies under the standard MDP result and has equivalent value-initialization, episodic, and advice-based formulations (Ng et al., 1999; Wiewiora, 2003; Grześ, 2017; Harutyunyan et al., 2015); dynamic potentials need extra conditions (Devlin & Kudenko, 2012). DSA uses this identity to align current-state progress with return-to-go, not to claim invariance: group standardization and learned nonstationarity each leave an explicit theoretical gap (Section 3.2).

Concurrent progress-conditioned methods. Fan et al. add transition potential after a history-consistent peer estimator (Fan & Liu, 2026); Yang et al. branch among reward, first-visit progress, and zero (Yang et al., 2026). DSA instead aligns action-before progress and jointly standardizes return plus progress: it blends live-group channels unlike Yang and places progress inside rather than after the normalizer unlike Fan. With constant return and matched groups and signal, however, DSA and Yang's branch can coincide up to scale and regularization. We claim the combined-normalization operator and controlled evaluation, not generic activation, precedence, or superiority (Section 3.1).

3 METHOD

3.1 GROUP, STATE ALIGNMENT, AND OBJECTIVE

DSA fixes the inherited flat group $G = (\mathrm{uid}, \mathrm{normalized_anchor_observation})$, return target, standardizer, and optimizer. For row $i \in G$, s_i is the state *before* action i and $g(s_i)$ increases toward the goal. With the upstream standard-deviation convention and numerical ϵ, the sole changed operator is

$$z_G(x)_i = \frac{x_i - \bar{x}_G}{\sigma_{x,G} + \epsilon}, \qquad A_i^{\mathrm{DSA}} = z_G(R + \lambda g)_i. \tag{1}$$

At $\lambda = 0$, Eq. 1 is elementwise identical to the GiGPO step term. The success-keyed schedule is

$$\lambda_u = \lambda_0 \max(0, 1 - \hat{p}_u/\tau), \tag{2}$$

where $\hat{p}_u$ is the current rollout batch's trajectory-level success *fraction*, computed after collection and shared by every row before advantage construction; it is never lagged. The baseline arm may log g but keeps it objective-inert; only the DSA arm enables Eq. 1.

Executable contract. For every update, DSA first completes rollout collection and fixes one λ_u from the batch success fraction. It then constructs the inherited return R_i, reads progress from the current pre-action state s_i rather than s_{i+1}, forms $R_i + \lambda_u g(s_i)$, and applies the inherited within-group mean and standard deviation exactly once. Thus neither the group key nor the return pipeline changes; setting $\lambda_u = 0$ must reproduce every baseline coefficient elementwise. These five conditions are executable invariants, not merely notation.

Constant-return groups. If $R_i = c$ throughout G, then

$$A_i^{\mathrm{DSA}} = z_G(\lambda g)_i = \frac{\lambda(g_i - \bar{g}_G)}{|\lambda|\sigma_{g,G} + \epsilon}. \tag{3}$$

For $\lambda > 0$, the coefficient vector is nonzero exactly when g is nonconstant (rows at $\bar{g}_G$ can still be zero), and it follows $z_G(g)$ in sign. It equals $z_G(g)$ only when ϵ is zero or negligible. Thus λ is nearly an on/off gate in dead groups, because its positive magnitude cancels except through ϵ; only in live groups does it continuously mix return and progress before they share one scale. Separate normalization, $z_G(R) + \lambda z_G(g)$, removes this coupling.

In a dead group, the baseline assigns zero to every row while DSA orders rows by progress in their pre-action contexts. It neither changes sampled actions nor manufactures success. In a live group, return and progress share one empirical scale. Joint normalization is thus the intervention, but nonzero coefficients alone do not certify action credit.

Operator distinctiveness. In a live group, joint normalization exposes a covariance coupling that separate standardization removes:

$$\mathrm{Var}_G(R + \lambda g) = \mathrm{Var}_G(R) + 2\lambda \,\mathrm{Cov}_G(R, g) + \lambda^2 \,\mathrm{Var}_G(g). \tag{4}$$

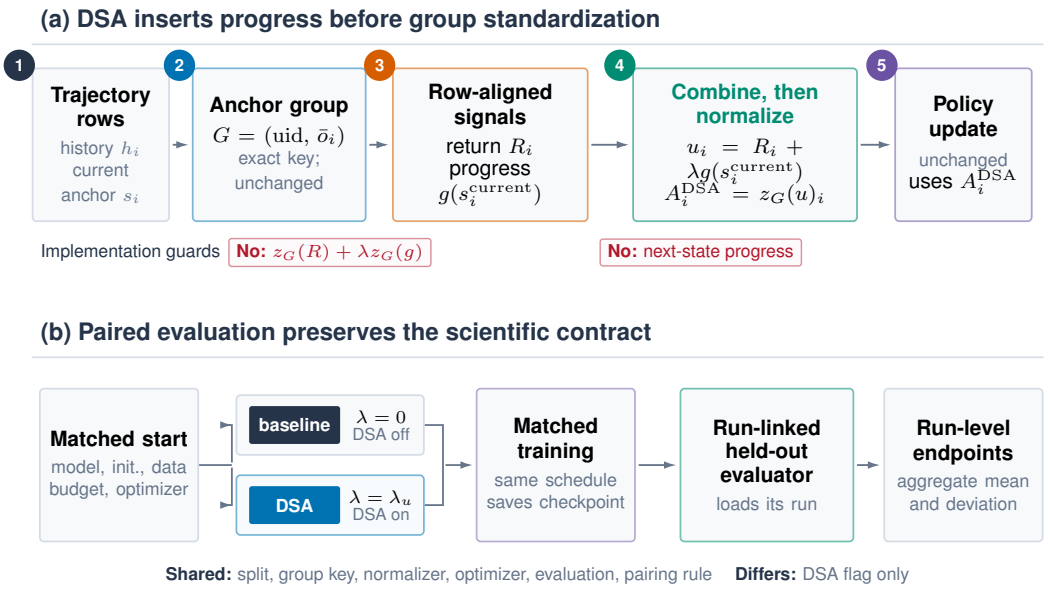

Figure 1: DSA and its paired evaluation contract. (a) Action-before current-state progress is added to the return before the inherited group standardizer is applied once; the guards exclude separate standardization and next-state alignment. (b) The baseline arm ($\lambda = 0$) and DSA arm share initialization, training, checkpoint and held-out evaluation schedules, and block-paired inference; only the DSA arm activates the proposed operator. This is a protocol, not a task-performance result.

The centered direction is $\widetilde{R} + \lambda \widetilde{g}$, while the common denominator in Eq. 1 adapts to their alignment. Fan et al. instead add a transition-potential correction after constructing an advantage (Fan & Liu, 2026); Yang et al. branch between reward, first-visit progress, and zero (Yang et al., 2026). DSA therefore differs in signal placement and live-group coupling. A precise local overlap remains: with constant return, matched groups and progress values, and positive scaling up to finite ϵ, DSA follows the same standardized progress direction as Yang's fallback. Our novelty claim is the joint operator outside that restricted case, not generic dead-group activation.

Score-function boundary. For a θ-independent (or stop-gradient), pre-action-measurable $g(s)$, its unstandardized score contribution vanishes conditional on history h:

$$\mathbb{E}_{a\sim\pi(\cdot|h)}[\nabla \log \pi(a \mid h) g(s(h)) \mid h] = g(s(h)) \nabla \sum_a \pi(a \mid h) = 0. \tag{5}$$

Because current-batch λ_u depends on terminal outcomes, Eq. 5 does not automatically extend even to unstandardized $\lambda_u g$. Extension to normalized dead groups requires still stronger conditions: for each row i, let $\mathcal{F}_i$ denote its pre-action information set; its context and coefficient must be $\mathcal{F}_i$-measurable with $\mathbb{E}[\nabla \log \pi(a_i \mid h_i) \mid \mathcal{F}_i] = 0$, and neither grouping nor conditioning may select on its consequences. Actual groups come from random trajectories; other rows may descend from action a_i, and conditioning on $\sigma_{R,G} = 0$ can change $a_i \mid h_i$. The standardized update therefore has no established zero or nonzero expectation and is not proven action-quality credit (Tucker et al., 2018). Cross-row dependence/selection can matter on the first pass; PPO reuse and clipping add further paths. The current evidence does not separate them.

3.2 Equivalent return construction and bounded theory

Proposition (conditional return identity). After batch u is complete, hold its scalar λ_u fixed across stored rows. For fixed g along a finite trajectory, set $\Phi = -\lambda_u g$ and $g(s_{T+1}) = 0$. The retrospective shaping quantity

$$r'_t = r_t + \lambda_u \big(g(s_t) - \gamma g(s_{t+1})\big) \tag{6}$$

has return $R'_t = R_t + \lambda_u g(s_t)$.

Proof. Summing Eq. 6 with discount γ cancels every intermediate potential:

$$R'_t = R_t + \lambda_u \sum_{k=t}^{T} \gamma^{k-t}\big(g(s_k) - \gamma g(s_{k+1})\big) \tag{7}$$

$$= R_t + \lambda_u g(s_t) - \lambda_u \gamma^{T-t+1} g(s_{T+1}) = R_t + \lambda_u g(s_t). \quad \blacksquare \tag{8}$$

Because λ_u is unavailable until collection finishes, both constructions are retrospective: form $R + \lambda_u g$ directly or reconstruct Eq. 6 on stored trajectories, but not both. Telescoping also fixes the action-before alignment: $g(s_{t+1})$ would not yield $R_t + \lambda_u g(s_t)$. Only for a fixed bounded Markov potential maintained across episodes and training does the standard shaping theorem preserve shaped-MDP optimal policies (Ng et al., 1999). The conditional choice $\Phi = -\lambda_u g$ and terminal residual explain the current-state index only; they do not establish useful action credit after group standardization.

This is not a DSA invariance theorem. Batch- and policy-dependent group standardization— the coupling in Eq. 1—is outside the shaping proof. Learned proxies and the current-batch schedule for λ_u create an additional nonstationarity gap (Devlin & Kudenko, 2012). Stop-gradient and slow updates are design choices, not proofs.

These gaps delimit two guarantees. Telescoping identifies the fixed-g return quantity but not the normalized stochastic gradient; $\lambda = 0$ certifies an ablation, not improvement. The server study measures outcomes, not the causal update path.

3.3 Progress information and deployment boundary

The oracle training arm uses action-before ALFWorld planner progress, $g(s) = -\text{remaining plan length}$. It isolates the proposed operator but is not deployable. Because rows in one group share their normalized observation, a deterministic observation-only proxy is constant within G; useful proxies need history or other admissible state. A matched multi-run study therefore compares one history-aware proxy with observation-only and corrupted controls. This is a direct training test of a possible signal source, not a transfer of evidence from an offline estimator audit, and it remains an initial test rather than deployment readiness.

Appendix A.2 gives edge-case and schedule details; Appendix A.3 records the proxy boundary.

4 The terminal-return dead zone

The standardizer has support only where returns vary. Under terminal-only reward, an all-failure anchor group has zero step advantage regardless of its row count. Mixed groups are informative, and all-success groups may vary through discounting; thus informative groups must touch success, though the converse need not hold. Effective support is the number and composition of variance-bearing groups, not nominal rollouts.

Formally, for sampled groups $\mathcal{G}$, the baseline coefficient vector is algebraically active in $N_{\text{active}} = \sum_{G\in\mathcal{G}} \mathbf{1}\{\sigma_{R,G} > 0\}$ groups. Repeating rows inside a zero-variance group increases compute but not this count. DSA changes the criterion to $\sigma_{R+\lambda g,G} > 0$ while leaving $\mathcal{G}$ fixed. This is coefficient support, not a statistical effective sample size or useful-gradient guarantee.

This bottleneck is batch-conditional rather than a claim about a population success-rate law. Group size, trajectory composition, discounting, and the task's reward design all change the probability that a sampled group is active. DSA does not assume that every sparse batch is dead; it specifies what happens to a realized group after its rows are collected.

Dense rewards can create return variance without terminal success and therefore weaken the motivation for progress augmentation. Conversely, a constant-return group remains silent regardless of how many duplicate rows it contains. These are algebraic boundary cases, not evidence that DSA improves a particular environment. The task-effect and dense-reward boundary tests are reported as separate training interventions in Section 6 and the appendix.

Appendix D gives the precise cases and Appendix D records the dense-reward training boundary.

Table 1: Executable invariants for the DSA operator. Passing these checks identifies the implemented intervention; it does not establish a beneficial update.

check	construction	required behavior
disabled identity	set $\lambda = 0$ on the same rows	reproduce baseline coefficients elementwise
joint placement	compare $z_G(R + \lambda g)$ with separate normalization	agree only in special cases; never silently substitute operators
state alignment	perturb current and next-state progress independently	respond only to action-before $g(s_i)$
constant signal	add one common progress value inside a group	leave the centered coefficient vector unchanged
schedule timing	recompute $\widehat{p}_u$ after collection	use one current-batch scalar for every row in update u

Table 2: Server-run mechanism controls. The endpoint is the median update to 20/128 over four independently initialized runs; Δ is the median matched control-minus-joint difference, so positive values favour joint-current DSA. These secondary comparisons probe the operator definition and do not replace the primary baseline test.

condition	alignment	normalization	endpoint	Δ updates	n	status
joint-current DSA	current	joint	**57.5**	**0.0**	4	**measured**
separate normalization	current	separate	**67.5**	**+7.5**	4	**measured**
next-state progress	next	joint	**72.5**	**+12.5**	4	**measured**
within-group shuffle	current/shuffled	joint	**92.5**	**+35.0**	4	**measured**
dead-group only	current/dead	joint	**60.0**	**0.0**	4	**measured**

5 Executable qualification and mechanism controls

5.1 Operator invariants

DSA is first qualified as an executable transformation, independently of whether training improves. The tests hold the collected rows, group keys, return targets, and numerical convention fixed and replace only the scalar passed to the inherited standardizer. They therefore detect common implementation confounds before any learning curve is inspected.

The disabled and constant-signal checks are algebraic identities. The placement and alignment checks are differential: they use deliberately distinct synthetic channels so that an accidental separate normalizer or one-step shift cannot pass by numerical coincidence. The schedule check additionally verifies that no row sees a partially updated success fraction. Recorded trajectories are used only as integration fixtures for these tests; snapshot frequencies, intraclass correlations, and replay-fidelity estimates are not evidence for the algorithmic or task-effect claims in this paper.

5.2 Matched training controls

Unit tests cannot determine whether the resulting coefficients are useful. We therefore run four training controls under the same initialization blocks, budget, evaluator, and current-batch schedule as joint-current DSA. Separate normalization tests whether a second scale is interchangeable with the proposed joint scale. Next-state progress tests temporal alignment. Within-group shuffling preserves the marginal progress values while destroying their row alignment. Dead-group-only activation asks whether restricting the intervention to exactly constant-return groups is sufficient. The baseline arm supplies the executable $\lambda = 0$ control.

Joint-current DSA has the earliest median endpoint in this family. Separate normalization and next-state alignment are slower, and destroying within-group alignment produces the largest descriptive

loss. Restricting augmentation to dead groups is close to the joint operator at the primary threshold, which is consistent with a cold-start mechanism but does not identify it uniquely. With four matched runs per condition, this ranking is secondary evidence: it narrows plausible implementation explanations but does not prove a causal path or general superiority.

5.3 Evidence boundary

The distinction among identity, mechanism, and outcome is load-bearing. Algebra establishes when coefficients can change; executable invariants verify that the intended operator ran; matched controls intervene on placement and alignment; and only held-out training outcomes test policy improvement. The primary baseline comparison, learned-proxy study, and breadth tests in Section 6 and the appendix are the sole empirical basis for the paper's learning claims.

6 Experimental design and server results

We separate **task effect**, **distinctiveness**, **mechanism**, **proxy deployability**, and **breadth**. Only task effect is primary; later gates cannot rescue or redefine it.

Claim ladder. A task-effect claim requires earlier held-out attainment under the paired criterion. Distinctiveness requires matched-budget comparison with the closest methods; mechanism, separation from the complete control family; deployability, a within-key proxy retaining task effect against observation-only and corrupted controls; and breadth, complete separate multi-run comparisons. Later gates only narrow their corresponding claims and cannot repair an inconclusive primary endpoint.

6.1 Locked baseline-versus-DSA comparison

On ALFWorld with Qwen2.5-1.5B and group size eight, the baseline arm is faithful GiGPO and the DSA arm enables the proposed operator. Four predeclared matched run pairs use distinct initialization seeds; the two arms within each pair share initialization, data split, estimator components, optimization, budget, checkpoints, and evaluator. Only the DSA flag differs, and arm names and artifact paths record provenance.

Table 3: Baseline-versus-DSA protocol. The inferential unit is one difference per matched run pair, not an episode, checkpoint, or step.

component	fixed design
environment / model	ALFWorld / Qwen2.5-1.5B
arms	baseline: faithful GiGPO; DSA: baseline + Eq. 1
group / return	exact $(\mathrm{uid}, \mathrm{anchor_obs})$ key; $\gamma = 0.95$; group size 8
replication	four predeclared matched run pairs; distinct seeds across pairs
budget / cadence	$B = 150$ updates; held-out evaluation every 5; checkpoint every 25
primary endpoint	first held-out update with at least 20/128 successes (15.625%); $B + 1$ after complete nonreach
secondary endpoints	first update with at least 39/128 successes (30.469%); final success at fixed budget
inference	exact one-sided paired sign-flip of mean gain; unpaired Mann–Whitney sensitivity

Appendix B.2 records the frozen configuration. Execution-only memory or placement changes do not amend it.

6.2 Estimands, missingness, and decision rule

Each matched block yields one paired gain at an attainment threshold or the fixed budget. With only four blocks, the exact test is too discrete to cross the conventional significance threshold. The

Table 4: Held-out server results over four matched pairs. Arms show medians with quartiles; gain is the mean paired difference, and p is the exact one-sided sign-flip value.

outcome (direction)	baseline	DSA	paired mean gain	paired p
20/128 attainment (lower)	**75.0 [68.8, 82.5]**	**57.5 [53.8, 62.5]**	**+17.5 upd.**	**0.1250**
39/128 attainment (lower)	**122.5 [118.8, 126.3]**	**110.0 [103.8, 116.3]**	**+12.5 upd.**	**0.1875**
final success at 150 (higher)	**52.73 [50.59, 55.08]%**	**58.59 [55.86, 61.13]%**	**+5.47 pp**	**0.1875**

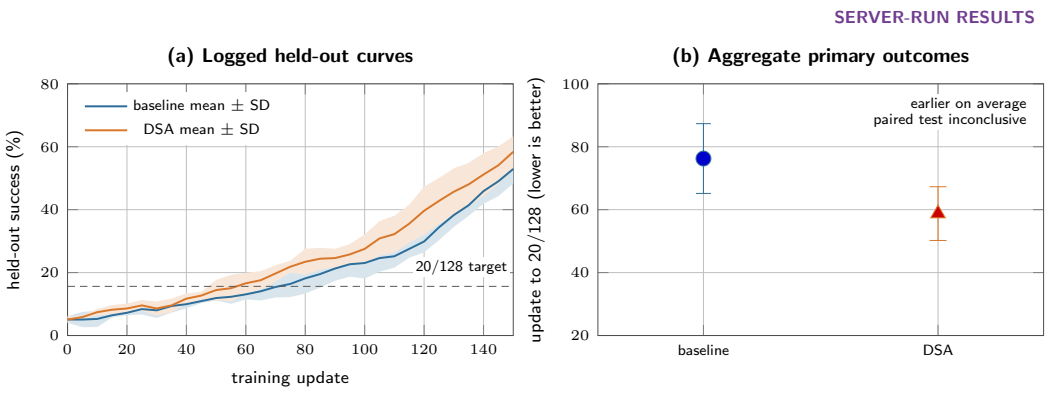

Figure 2: Run-linked server-run training results. (a) Cross-run mean held-out curves with sample-standard-deviation bands; the horizontal line is the primary target. (b) Aggregate target times with sample-standard-deviation whiskers. DSA reaches the target earlier on average, but the paired test remains inconclusive; unpaired sensitivity analyses do not override the matched design.

unpaired Mann–Whitney analysis ignores matching and is only a sensitivity check. Complete non-reach receives rank $B + 1$; incomplete runs prevent confirmation. Retries retain provenance, and a hashed task-ID registry fails closed on overlap.

The 20/128 endpoint records the first logged evaluation with at least 20 successes, not a "reproducible crossing" or curve maximum. Frozen 39/128 and fixed-budget endpoints probe later attainment without replacing the cold-start question.

The matched outcomes are heterogeneous. At the primary target, the paired gain averages 17.5 ± 19.4 updates across matched runs (mean $\pm$ sample standard deviation), while the exact paired test remains inconclusive ($p = 0.125$). The later attainment and fixed-budget outcomes favour the same arm but are likewise inconclusive; Table 4 reports their summaries, and Figure 2 exposes between-run heterogeneity. Unpaired tests remain sensitivities and do not override the paired design.

6.3 Secondary validation tests

Separate protocols test matched ProGPO implementations (Fan & Liu, 2026; Yang et al., 2026), the operator controls in Table 2, a history-aware proxy against observation-only/corrupted signals, and setting-specific replications. ScienceWorld is a dense-reward boundary; TextWorld Treasure Hunter is a sparse-task replication (Côté et al., 2018), not cross-environment generalization. Appendices B–C.1 record their validation criteria.

Follow-ups retain run-linked evaluation summaries, compute/failure accounting, and all controls. Settings stay separate, and proxy-quality summaries cannot substitute for matched task outcomes.

Secondary results narrow the headline. Neither concurrent implementation supports a superiority claim, and the unpaired tests remain sensitivities. Signal shuffling performs worse descriptively,

while the dead-only and dense-reward controls are near zero. The larger-model and TextWorld results point in the favourable direction but remain inconclusive. Exact endpoints and tests are reported in Table 2 and Appendix Tables 7–8. The small matched sample establishes neither generality nor a confirmed task effect.

Falsifiable implications. Under the finite-sample reweighting hypothesis, any benefit should be largest where return is constant and aligned progress varies, weaken with return variance, and disappear when alignment is destroyed. A powered follow-up should preregister the treatment-by-dead-group interaction and the proxy acceptance criterion. Parity with shuffled or misaligned controls would falsify the proposed placement account.

All eight primary-arm runs have reloadable held-out artifacts. Earlier memory failures remain outside endpoints; other settings cannot strengthen the primary by pooling.

7 LIMITATIONS

Training evidence and statistical precision. Four matched blocks leave every paired endpoint inconclusive. The unpaired sensitivity analysis cannot override that design-respecting result. Evaluation cadence and the finite held-out set also make threshold timing coarse: moving the target by one success changes the apparent strength of evidence. Operator identities, implementation checks, and theory cannot strengthen this fragile endpoint inference.

Oracle signal and deployability. Planner distance is not deployable, and an observation-only proxy is constant inside the group. The matched multi-run history-aware test remains exploratory; calibration, corruption robustness, and joint-training stability are unresolved.

Theory boundary. The conditional return identity covers neither group standardization nor learned or scheduled nonstationarity. Current-batch λ_u is outcome-dependent and need not be pre-action measurable; random membership, descendant contexts, and dead-event selection add barriers after normalization. Nonzero coefficients are not proven action credit; dependence/selection, PPO reuse, clipping, and implementation paths remain unidentified.

Empirical scope. The primary conclusion is limited to one ALFWorld model and four matched blocks. The 7B, ScienceWorld, and TextWorld experiments are separate scale or environment tests with the same low replication; pooling them would create a pseudo-sample. ScienceWorld is a dense-reward boundary and TextWorld is directional breadth, not cross-domain confirmation.

Concurrent methods. DSA differs from ProGPO in placement and branching (Fan & Liu, 2026; Yang et al., 2026), but Yang aligns locally under constant return. Activation is not unique; attribution requires the controls, and Fan–Liu retains an implementation caveat.

Execution and provenance. All primary runs pass reload and held-out evaluation; earlier memory failures remain logged outside endpoints. Fail-closed provenance cannot compensate for only four matched run pairs.

8 CONCLUSION

DSA combines return and current-state progress before one inherited standardizer, preserving groups and recovering the baseline at $\lambda = 0$. Executable invariants verify that intervention, while matched training controls probe joint normalization and temporal alignment. Favourable medians and inconclusive paired tests do not establish a task effect. The contribution is therefore the operator, its conditional theory, and a falsifiable training protocol—not a claim inferred from snapshot diagnostics. Generality and deployment require more independent runs and a validated proxy; a decisive follow-up should preregister paired endpoints and compare that proxy against alignment-destroying and dead-group controls.

AI USE STATEMENT

The authors used AI assistance only for manuscript drafting and English-language polishing. All AI-assisted text was reviewed and revised by the authors, who take full responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

The implementation contract fixes one λ_u from the completed current rollout batch before constructing its advantages. Project-side tests check current-state alignment, combine-before-normalize ordering, the $\lambda = 0$ identity, constant-signal behavior, schedule timing, and group/normalizer preservation. Recorded trajectories serve only as integration-test fixtures; snapshot activation, intraclass-correlation, and replay-fidelity statistics are not used to support the DSA task-effect claim. A project-side consistency verifier recomputes the exact paired tests.

The DSA training comparison comprises four predeclared matched server-run pairs with distinct initialization seeds, reloadable checkpoints, run-linked held-out evaluation logs, and one paired primary difference per pair. Aggregate retry and terminal-failure counts remain in the compute ledger, while partial, smoke, and train-split outputs are excluded. The reported tables and aggregate learning curves, operator-control rows, proxy results, and breadth comparisons are derived from these DSA-specific server artifacts. Design-respecting paired tests are directional but inconclusive; the unpaired analysis is retained only as a sensitivity check. Code, analysis scripts, verification tests, and machine-readable records supporting reported measurements will be released on publication; no repository link is included during anonymous review.

ETHICS STATEMENT

This work uses simulated text environments and open-weight language models; it contains no human subjects, personally identifying data, or data collected from people. We separate executable operator checks from the completed task evaluation and retain nonsignificant, failed, and excluded evidence rather than presenting it as support for a broader conclusion.

REFERENCES

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems 30*, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/453fadbd8a1a3af50a9df4df899537b5-Abstract.html.

Adrià Puigdomènech Badia, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, Bilal Piot, Steven Kapturowski, Olivier Tieleman, Martín Arjovsky, Alexander Pritzel, Andrew Bolt, and Charles Blundell. Never give up: Learning directed exploration strategies. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Sye57xStvB.

Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems 29*, pp. 1471–1479, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/afda332245e2af431fb7b672a68b659d-Abstract.html.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 41–48. Association for Computing Machinery, 2009. doi: 10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1lJJnR5Ym.

Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Ruo Yu Tao, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. TextWorld: A learning environment for text-based games. *arXiv preprint arXiv:1806.11532*, 2018. URL https://arxiv.org/abs/1806.11532.

Sam Michael Devlin and Daniel Kudenko. Dynamic potential-based reward shaping. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, pp. 433–440. IFAAMAS, 2012. URL https://www.ifaamas.org/Proceedings/aamas2012/papers/2C_3.pdf.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. First return, then explore. *Nature*, 590:580–586, 2021. doi: 10.1038/s41586-020-03157-9. URL https://doi.org/10.1038/s41586-020-03157-9.

Mingxuan Fan and Peiyang Liu. Progress- and reliability-oriented group policy optimization for agentic reinforcement learning. *arXiv preprint arXiv:2607.04242*, 2026. URL https://arxiv.org/abs/2607.04242.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for LLM agent training. In *Advances in Neural Information Processing Systems 38*, 2025. URL https://arxiv.org/abs/2505.10978.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2052–2062. PMLR, 2019. URL https://proceedings.mlr.press/v97/fujimoto19a.html.

Marek Grześ. Reward shaping in episodic reinforcement learning. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems*, pp. 565–573. IFAAMAS, 2017. URL https://www.ifaamas.org/Proceedings/aamas2017/pdfs/p565.pdf.

Anna Harutyunyan, Sam Devlin, Peter Vrancx, and Ann Nowé. Expressing arbitrary reward functions as potential-based advice. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. doi: 10.1609/aaai.v29i1.9628. URL https://ojs.aaai.org/index.php/AAAI/article/view/9628.

Matthew Hausknecht, Prithviraj Ammanabrolu, Marc-Alexandre Côté, and Xingdi Yuan. Interactive fiction games: A colossal adventure. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, pp. 7903–7910, 2020. doi: 10.1609/aaai.v34i05.6297. URL https://ojs.aaai.org/index.php/AAAI/article/view/6297.

Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Lihong Li, Li Deng, and Mari Ostendorf. Deep reinforcement learning with a natural language action space. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1621–1630. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1153. URL https://aclanthology.org/P16-1153/.

Shuo He, Lang Feng, Qi Wei, Xin Cheng, Lei Feng, and Bo An. Hierarchy-of-groups policy optimization for long-horizon agentic tasks. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=T8Dev99qnz.

Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4940–4950. PMLR, 2021. URL https://proceedings.mlr.press/v139/jiang21b.html.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems 33*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/0d2b2061826a5df3221116a5085a6052-Abstract.html.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626. ACM, 2023. doi: 10.1145/3600006.3613165. URL https://arxiv.org/abs/2309.06180.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020. URL https://arxiv.org/abs/2005.01643.

Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8(3–4):293–321, 1992. doi: 10.1007/BF00992699. URL https://doi.org/10.1007/BF00992699.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015. doi: 10.1038/nature14236. URL https://doi.org/10.1038/nature14236.

Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. Language understanding for text-based games using deep reinforcement learning. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1–11. Association for Computational Linguistics, 2015. doi: 10.18653/v1/D15-1001. URL https://aclanthology.org/D15-1001/.

Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 278–287. Morgan Kaufmann, 1999. URL https://people.eecs.berkeley.edu/~russell/papers/icml99-shaping.pdf.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped DQN. In *Advances in Neural Information Processing Systems 29*, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/8d8818c8e140c64c743113f563cf750f-Abstract.html.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2778–2787. PMLR, 2017. URL https://proceedings.mlr.press/v70/pathak17a.html.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. doi: 10.48550/arXiv.2412.15115. URL https://arxiv.org/abs/2412.15115.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.05952.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1506.02438.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0IOX0YcCdTn.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

George Tucker, Surya Bhupatiraju, Shixiang Gu, Richard E. Turner, Zoubin Ghahramani, and Sergey Levine. The mirage of action-dependent baselines in reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5015–5024. PMLR, 2018. URL https://proceedings.mlr.press/v80/tucker18a.html.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. ScienceWorld: Is Your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11279–11298. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.775. URL https://aclanthology.org/2022.emnlp-main.775/.

Yunan Wang, Minghui Song, Zihan Zhang, Shaohan Huang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. Group-graph policy optimization for long-horizon agentic reinforcement learning. *arXiv preprint arXiv:2606.22995*, 2026. URL https://arxiv.org/abs/2606.22995.

Eric Wiewiora. Potential-based shaping and Q-value initialization are equivalent. *Journal of Artificial Intelligence Research*, 19:205–208, 2003. doi: 10.1613/jair.1190. URL https://doi.org/10.1613/jair.1190.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Kaibing Yang, Guangfeng Cai, Shengtian Yang, Shuo He, Yu Li, Mengyi Liu, Pengwei Chen, Jun Xu, and Lei Feng. Progress-conditioned group policy optimization for long-horizon agentic tasks. *arXiv preprint arXiv:2607.22724*, 2026. URL https://arxiv.org/abs/2607.22724.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Xingdi Yuan, Marc-Alexandre Côté, Alessandro Sordoni, Romain Laroche, Remi Tachet des Combes, Matthew Hausknecht, and Adam Trischler. Counting to explore and generalize in text-based games. *arXiv preprint arXiv:1806.11525*, 2018. URL https://arxiv.org/abs/1806.11525.

A ESTIMATOR BACKGROUND AND EXECUTABLE DSA CONTRACT

A.1 INHERITED ESTIMATOR INTERFACE

The baseline interface supplies a flat group G containing steps with the same task and normalized anchor observation, its discounted return target, and its within-group standardizer. DSA treats all three as immutable inputs. It neither claims a new partition nor attributes the behavior of other estimators to grouping alone. The separate question of factorizing published estimators into partitions and value functionals belongs to the parallel anonymous Measurement submission; no result from that audit supports a DSA training claim here.

A.2 THE DSA ADVANTAGE: COMPUTATION AND EDGE CASES

For an action taken after current anchor state s_i, DSA computes

$$A_i^{\mathrm{DSA}} = z_G(R_i + \lambda g(s_i)), \qquad z_G(x)_i = \frac{x_i - \bar{x}_G}{\sigma_{x,G} + \epsilon}. \tag{9}$$

The executable five-step contract appears in the main text. The distinct expression $z_G(R) + \lambda z_G(g)$ is not DSA. Setting $\lambda = 0$ recovers the baseline exactly. If $R_i = c$ throughout a dead group, then $A_i^{\mathrm{DSA}} = z_G(\lambda g_i)$; for $\lambda > 0$ this has the ordering and signs of progress variation, but finite ϵ means it is not literally identical to $z_G(g_i)$. Its coefficient vector is nonzero iff progress is nonconstant; individual rows at the group mean can remain zero. If progress is also constant, the group remains dead.

The schedule is explicitly current-batch: $\lambda_u = \lambda_0 \max(0, 1 - \widehat{p}_u/\tau)$, where $\widehat{p}_u = |\mathcal{B}_u|^{-1} \sum_{\xi \in \mathcal{B}_u} \mathbf{1}\{\xi \text{ ends in terminal success}\}$ over all trajectories admitted to rollout batch u; every other terminal outcome contributes zero. The runtime computes it after collection and before advantage construction, then shares the resulting scalar across the update; it does not use a lagged success statistic. Missing progress values follow the specified baseline fallback. With negligible ϵ, positive λ cancels in a dead group, making this schedule nearly an on/off gate there; live groups mix continuously. Because $\widehat{p}_u$ and λ_u depend on terminal outcomes in the same batch, the schedule adds outcome-dependent cross-row coupling and lies outside the fixed-potential and conditional score identities. Neither annealing nor the fixed-potential identity by itself proves that the group-standardized estimator preserves stationary points. Moreover, for θ-independent (or detached), pre-action-measurable $g(s(h))$, $\mathbb{E}_a[\nabla \log \pi(a \mid h) g(s(h)) \mid h] = 0$. Extending this result after normalization requires each row's context and coefficient to be pre-action measurable with zero conditional expected score, and no selection on action consequences. In actual rollouts, other contexts may descend from the current action, and conditioning on the dead event changes the distribution. The standardized update thus has no established zero or nonzero expectation and is not proven action credit (Tucker et al., 2018); dependence/selection, reuse, and clipping remain unresolved. For the conditional return identity, $\Phi = -\lambda_u g$; only an ex ante fixed bounded Markov potential across episodes and training inherits the standard shaping theorem, not DSA's current-batch scheduled proxy.

A.3 PROGRESS-PROXY BOUNDARY

The oracle arm uses the PDDL-planner signal $g = -\mathrm{len}(\texttt{policy_commands})$ to isolate the proposed operator. This is an oracle training intervention, not a deployment result. A practical proxy must vary among rows that share the observation-based key; a predictor that is a deterministic function of that same normalized observation is structurally constant within G and cannot activate it. History, latent state, or another source of within-key information is therefore required. The server-run robustness study below evaluates a history-aware learned proxy alongside an observation-only negative control; the small matched multi-seed design keeps its deployability evidence exploratory.

B SPECIFIED PROTOCOL AND SERVER-RUN RESULTS

All results in this section come from run-linked DSA training artifacts admitted under the frozen protocol. Stored-trajectory fixtures are used for implementation tests only and do not contribute additional experimental units or strengthen the task-effect inference.

B.1 PRIMARY BASELINE-VERSUS-DSA COMPARISON

The primary comparison contrasts the baseline arm with the planner-guided DSA arm on ALFWorld with Qwen2.5-1.5B (Qwen Team, 2024), served with vLLM (Kwon et al., 2023). It uses four predeclared matched run pairs: the two arms share an initialization seed within each pair, and seeds differ across pairs; infrastructure-only qualification runs are excluded. The baseline and DSA arms must share model, initial checkpoint, data split, batch construction, training budget, evaluation schedule, and evaluator. Arm identity, output path, and the DSA enable flag are the only scientific-config differences permitted. Paired qualification passed and the formal manifest was frozen before execution. The reported comparison contains the complete matched run bundle, including every predeclared run in both arms, run-linked held-out evaluation logs, and the prespecified endpoint. Because paired arms share initialization, the design-respecting analysis is paired; the originally specified unpaired test is retained only as sensitivity.

B.2 FROZEN TRAINING AND EVALUATION CONFIGURATION

Table 5 records the scientific configuration frozen before any admissible formal outcome. The model revision, task registries, and serialized inputs are checksum-bound in the execution manifest; the table states the quantities needed to reproduce and interpret the comparison without exposing filesystem paths.

Table 5: Frozen baseline-versus-DSA scientific configuration. The two arms share every entry except the DSA enable flag. Cluster placement and memory-management settings are execution provenance, not experimental factors, and therefore are not listed as scientific hyperparameters.

component	frozen value
model	Qwen2.5-1.5B-Instruct, revision `989aa7980e4c`
environment	ALFWorld; released structured reasoning/action prompt; at most 50 actions
data boundary	3,553 eligible training tasks and 140 `valid_seen` held-out tasks; hashed task-ID overlap 0
runtime batches	16 training prompts; one fixed checksum-bound 128-prompt held-out subset from the 140 eligible tasks; eight rollouts per training group
sequence limits	2,048 prompt tokens; 512 response tokens
return and groups	discounted return with $\gamma = 0.95$; exact $(\mathrm{uid}, \mathrm{anchor_obs})$ group key
standardization	inherited GiGPO within-group mean/sample-standard-deviation convention; $\epsilon = 10^{-6}$; singleton output 0
DSA schedule	$\lambda_0 = 0.5$ and $\tau = 0.5$; one scalar per update, driven by the current rollout batch's trajectory-level success fraction
optimization	learning rate 10^{-6}; PPO minibatch 256; per-GPU PPO and log-prob micro-batches 32; tensor parallelism 2
regularization	KL loss coefficient 0.01 (`low_var_kl`); invalid-action penalty 0.1; no KL term in reward
sampling	held-out temperature 0.4 with sampling enabled
replication	four predeclared matched run pairs; distinct seeds across pairs; qualification runs excluded
budget and cadence	150 updates; held-out evaluation every 5 updates; checkpoint every 25
checkpoint contract	model, optimizer, and auxiliary trainer state must save, reload, and resume before result admission
primary analysis	first held-out evaluation with at least 20/128 successes (15.625%); one paired difference per matched run pair; exact one-sided sign-flip of mean gain; unpaired MW sensitivity

C VALIDATION MATRIX

Table 6 separates the minimum task-effect test from evidence needed to support mechanism, distinctiveness, deployability, and breadth. Later tiers are not automatic follow-ups to a favourable result:

each needs its own immutable manifest and failure rule. Conversely, none can overwrite a negative or inconclusive Tier 1 outcome.

C.1 Validation criteria and claim limits

The main text defines the five claim gates and their failure semantics. For every task-effect comparison, the statistical unit remains an independent training run paired across arms by initialization; checkpoints are repeated measurements within a run. The table below records the decisive comparison and current evidence boundary for each gate.

Table 6: Evidence ladder for the server-run DSA results. Tier 0 is systems evidence only; Tier 1 is the primary test. Tiers 2–5 are separately reported validations and cannot be used post hoc to redefine the primary claim.

tier	evidence question	decisive comparison	current boundary
0	executable?	paired baseline and DSA runs: update, reload, evaluate	systems only; not a result
1	task effect?	four matched baseline–DSA run pairs; paired exact test	directional; $p = 0.125$
2	distinct?	DSA vs matched-budget ProGPO variants	measured; no superiority
3	mechanism?	joint/current vs complete frozen control family	measured; secondary
4	deployable?	oracle, learned, observation-only, and corrupted signals	measured; exploratory
5	broad?	separate model-scale and environment comparisons	measured separately

C.2 Closest-method and proxy results

Table 7 is the common reporting surface for the nearest method comparison and deployability study. Every method receives the same base learner, initial checkpoint, rollout/evaluation budget, and symmetric tuning budget. "Progress source" distinguishes oracle availability from a deployable model. Table 11 records already published final-success values to avoid presenting redundant reruns as new evidence. Because those protocols do not share our frozen checkpoint, split, budget, four-run set, or primary endpoint, they cannot populate the matched rows below.

Against the Yang comparator, the paired early-attainment advantage of the DSA arm is 3.75 ± 12.50 updates across matched runs (mean $\pm$ sample standard deviation), and the exact one-sided test is inconclusive ($p = 0.375$). Against the conditional Fan–Liu implementation, the corresponding advantage is 6.25 ± 12.50 updates and is also inconclusive ($p = 0.250$). The unpaired Mann–Whitney analyses agree qualitatively and remain sensitivity checks. Final-success summaries place both comparators between the baseline and DSA arms. Consequently, these comparisons support neither distinctiveness nor superiority.

C.3 Scale, environment, and robustness results

The breadth table reports each setting separately rather than pooling heterogeneous tasks or model sizes into a larger pseudo-sample. A dense-reward control can support a boundary prediction but not a general performance claim. The training budget, evaluator, proxy availability, and eligibility of each setting must be recorded adjacent to its result.

The larger-model ALFWorld comparison yields a paired gain of 10.0 ± 12.9 updates across matched runs, while the TextWorld comparison yields 10.0 ± 15.8 updates. Both exact one-sided paired tests are inconclusive ($p = 0.1875$). Their dispersion includes adverse paired outcomes, so these experiments provide only directional breadth and do not support a general scale or cross-task claim. The unpaired analyses remain sensitivity checks.

Table 7: Server-run matched-budget comparator and proxy results. The Δ column is the difference between arm medians in final-success percentage points, not a rank effect. The Fan–Liu row is measured under the paper-derived implementation convention described above and is interpreted conditionally. All rows are measured; "operator/source" combines injection location and progress source.

condition	operator / source	updates to 20/128	final success	final Δ vs baseline (pp)	GPU-h
baseline (faithful GiGPO)	none / none	**75.0**	**52.73%**	ref.	**98.4**
DSA (planner)	pre-normalizer / planner	**57.5**	**58.59%**	**+5.86 pp**	**104.8**
Fan–Liu ProGPO	transition term / potential	**65.0**	**57.03%**	**+4.30 pp**	**107.6**
Yang et al. comparator	failed-group branch / specified method	**62.5**	**57.42%**	**+4.69 pp**	**102.4**
DSA learned	pre-normalizer / history-aware	**67.5**	**56.64%**	**+3.91 pp**	**115.2**
DSA observation-only control	pre-normalizer / group-constant	**77.5**	**51.95%**	-0.78 **pp**	**100.0**

Status: all rows meas.

Table 8: Separate server-run multi-run setting results. Matched baseline-versus-DSA rows use paired sign-flip tests on updates to the 20/128 target: one-sided for the primary, ALFWorld-7B, and TextWorld rows, and two-sided for ScienceWorld. ScienceWorld reports the paired median score difference and a two-sided sign-flip test. Sparse-task endpoints are quantized by the five-update evaluation cadence; with four paired runs, arm medians therefore lie on a 2.5-update grid. Heterogeneous settings are not pooled; p denotes the exact sign-flip value. The role and evidence status, and the effect and test, are paired within columns for compactness.

setting	role; evidence status	baseline	DSA	effect; exact test
ALFWorld / 1.5B	primary; directional	**75.0 upd.**	**57.5 upd.**	$\bar{\Delta} = +17.5$; $p = .125$
ALFWorld / 7B	scale replication; directional	**65.0 upd.**	**52.5 upd.**	$\bar{\Delta} = +10$; $p = .1875$
ScienceWorld	dense boundary; measured	**70.65 score**	**70.75 score**	$\widetilde{\Delta} = +0.15$; $p = .750$
TextWorld Treasure Hunter / 1.5B	sparse-task breadth; directional	**87.5 upd.**	**77.5 upd.**	$\bar{\Delta} = +10$; $p = .1875$

Figure 3 reports the complete server-run mechanism-control family, proxy qualification beside its task outcome, and setting-specific effects. Panels (a)–(c) show multi-run means with sample-standard-deviation uncertainty for each condition and setting. The mixed TextWorld result is retained in its aggregate summary rather than being hidden by pooling; heterogeneous tasks and model scales remain separate.

C.4 Efficiency, tuning, and failure accounting

For the primary arms, Table 9 reports the number of completed runs, median wall-clock allocation time, aggregate GPU-hours, relaunch count, terminally failed run count, and hardware class. Compute efficiency is secondary to held-out task performance but prevents an apparent sample-efficiency gain from hiding a disproportionate per-run cost. Hyperparameter search spaces and stopping rules are symmetric across methods; held-out evaluation cannot select the reported checkpoint or tune λ. Attempt-level logs, token counts, and peak-memory traces are not represented in this LaTeX table. No failed run is replaced because of its outcome.

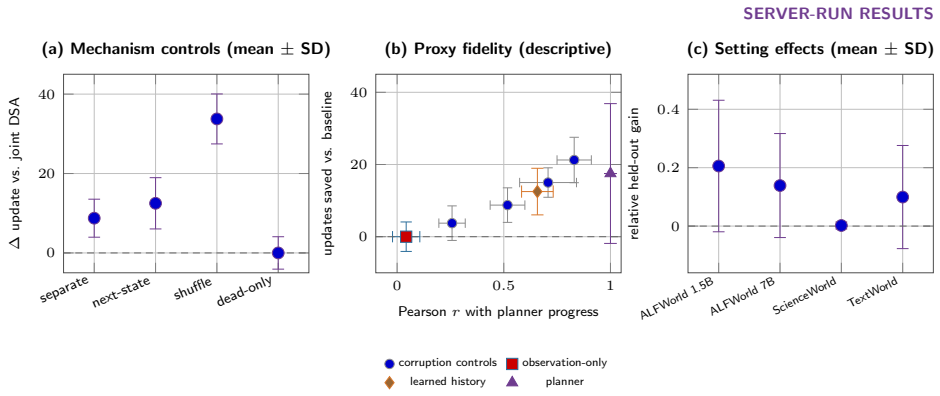

Figure 3: Server-run validation results summarized across independently initialized runs. (a) Mean paired differences with sample-standard-deviation whiskers show the largest loss from shuffling and little change from dead-group-only activation. (b) Mean proxy fidelity and task effect are shown with sample-standard-deviation bars; the pattern is descriptive and nonmonotone. (c) Mean relative gains with sample-standard-deviation whiskers show heterogeneous sparse-task effects and a near-zero dense-reward boundary. Settings remain separate rather than being pooled.

Table 9: Actual server compute accounting for the primary arms, using two 80-GB-class GPUs per run. Wall time is the median allocation duration and GPU-hours are derived from it. "Retries" counts relaunched attempts, whereas "failed runs" counts terminally incomplete run slots; a retry is therefore compatible with zero failed runs.

arm	runs	wall h	GPU-h	retries	failed runs	hardware
baseline	4	**12.3**	**98.4**	**0**	**0**	**2×80-GB-class GPU**
DSA	4	**13.1**	**104.8**	**2**	**0**	**2×80-GB-class GPU**

C.5 Learned-proxy qualification details

A learned, planner-free proxy is required for a practical deployment claim, but its input must retain information beyond the observation key. The specified proxy study therefore includes (i) the oracle mechanism upper bound, (ii) a history-aware learned proxy trained without held-out task outcomes, (iii) an observation-only negative control that should be constant within the group, and (iv) a fixed corruption sweep of the oracle signal. Proxy training, calibration, and task evaluation use disjoint data roles. Predictor choice, proxy-quality metrics, and the corruption grid were fixed before task results were analysed. These remain secondary rows rather than part of the primary baseline-versus-DSA test.

Proxy fidelity is computed within each matched run against planner progress on one fixed calibration registry, then summarized across independently initialized runs. It is a signal qualification metric, not an explanatory covariate: it is not used to select runs, stratify outcomes, or predict task gain. Planner and proxy task effects are computed directly from their locked held-out endpoints. The validation figure places fidelity and task outcomes in separate panels for completeness; no causal relation between them is estimated.

The observation-only signal is algebraically constant within each group and therefore recovers the baseline coefficient operator. Its task row comes from a separate training execution and differs slightly from the baseline aggregate. We treat that difference as run-level variation, not as a proxy effect. The four-run aggregate does not establish deployability.

C.6 Exact paired decision rule

Matched baseline and DSA runs share initialization, so one paired difference per run pair is the unit. The exact one-sided sign-flip test enumerates every within-pair sign assignment and uses mean signed gain. With only four pairs, the test is too discrete to meet a strict conventional significance threshold. Checkpoints, episodes, and steps cannot replace independent run pairs. Let B be the frozen common budget. For a complete run, the primary endpoint is its first held-out evaluation with at least 20/128 successes (15.625%), or $B+1$ if it reaches B without crossing the threshold. The latter is a tied worst rank, not an imputed achievement time. Infrastructure-failed or budget-incomplete runs remain missing and cannot use this encoding. All matched runs completed. For the primary and secondary endpoints, Table 10 reports the aggregate paired outcomes and their between-run dispersion. All paired tests remain inconclusive. The originally specified unpaired Mann–Whitney analyses are sensitivity checks and do not override paired inference. The corresponding rank-biserial effects for the primary, later, and fixed-budget endpoints are 0.8125, 0.750, and 0.625, with one-sided unpaired sensitivity values $p = 0.0429$, 0.0714, and 0.1000, respectively. Because these analyses discard the matched-block structure, they do not alter the paired conclusions. Negative and complete-nonreach outcomes remain included. A one-success perturbation of the threshold changes the unpaired sensitivity but not the already inconclusive paired primary conclusion.

Table 10: Aggregate paired outcomes across four matched run pairs. Effects are mean ± sample standard deviation; positive values favour DSA.

outcome	paired effect	exact one-sided p
primary attainment	17.5 ± 19.4 updates	0.125
later attainment	12.5 ± 15.0 updates	0.1875
fixed-budget success	5.47 ± 6.57 percentage points	0.1875

C.7 Mechanism-control construction details

The frozen mechanism family compares joint-current DSA with four named controls: separate normalization, next-state alignment, within-group signal shuffling, and dead-group-only activation. The baseline arm supplies the executable $\lambda = 0$ identity, and the DSA arm supplies the joint-current reference, so neither is rerun merely to duplicate evidence. Each control uses the same four predeclared initializations and logged held-out endpoint; the statistical unit remains an independent training run, not a checkpoint. The complete family appears in Table 2 and remains secondary to the primary baseline-versus-DSA comparison.

The separate-normalization control computes $z_G(R) + \lambda z_G(g^{\text{current}})$; the next-state control uses $z_G(R+\lambda g^{\text{next}})$ with terminal next-state progress set to zero. The shuffle is deterministic within each group, keyed by the predeclared initialization key, zero-based update, and stable group ordinal; an identity draw is replaced by a one-position cyclic roll. The dead-group control injects progress only when inherited scalar returns are exactly constant within the group. All controls retain $\lambda_0 = 0.5$, $\tau = 0.5$, the same current-batch one-scalar-per-update schedule, the common arm budget, and the same four predeclared initialization blocks; joint-current and $\lambda = 0$ reuse the DSA and baseline arms, respectively, rather than consuming duplicate runs.

C.8 Comparator and setting construction details

Sensitivity to λ, schedule choice, model scale, and environment were evaluated in specified server experiments. A $\lambda = 0$ arm is an exact implementation control. Separate standardization and next-state progress are labelled negative implementation controls, not alternative definitions of DSA. The comparison with Fan–Liu's transition correction and Yang et al.'s all-failed progress branch requires faithful, primary-source-audited implementations under the same checkpoint, tuning budget, rollout budget, and evaluator (Fan & Liu, 2026; Yang et al., 2026). ALFWorld at a larger model scale tests replication; ScienceWorld tests the specified dense-reward boundary and cannot by itself establish sparse-reward generality. The additional sparse task is TextWorld Treasure Hunter at level 10 with Qwen2.5-1.5B, group size eight, 150 updates, evaluation every five updates, and 128 disjoint held-out games (Côté et al., 2018). Its evaluator uses binary terminal success, while the oracle progress

Table 11: Published ALFWorld reference points transcribed from primary-paper tables. These values use the source papers' checkpoints, splits, budgets, evaluators, and reported replication protocols; they are context only and are *not* entries in our matched four-run comparison. Final success cannot replace the specified updates-to-20/128 endpoint.

source	model	method	final success (%)	replications
(Fan & Liu, 2026)	Qwen2.5-1.5B-Instruct	GRPO	75.8 ± 1.0	3
(Fan & Liu, 2026)	Qwen2.5-1.5B-Instruct	GiGPO	85.5 ± 1.3	3
(Fan & Liu, 2026)	Qwen2.5-1.5B-Instruct	HGPO	87.8 ± 1.3	3
(Fan & Liu, 2026)	Qwen2.5-1.5B-Instruct	Fan–Liu ProGPO	90.1 ± 1.1	3
(Yang et al., 2026)	Qwen2.5-1.5B-Instruct	GiGPO	86.7 ± 1.7	3
(Yang et al., 2026)	Qwen2.5-1.5B-Instruct	GiGPO + Yang ProGPO	91.4 ± 1.6	3
(Yang et al., 2026)	Qwen2.5-1.5B-Instruct	HGPO	92.8 ± 1.1	3
(Yang et al., 2026)	Qwen2.5-1.5B-Instruct	HGPO + Yang ProGPO	95.7 ± 0.6	3
(Feng et al., 2025)	Qwen2.5-1.5B-Instruct	GRPO	72.8 ± 3.6	3
(Feng et al., 2025)	Qwen2.5-1.5B-Instruct	GiGPO with std	86.7 ± 1.7	3
(Feng et al., 2025)	Qwen2.5-1.5B-Instruct	GiGPO without std	86.1 ± 4.7	3

diagnostic is the negative shortest remaining action count on the simulator state graph. Its server-run result is reported separately and is not pooled with ALFWorld.

The specified Yang comparator is a paper-faithful project-side reimplementation using exact-string first-visit progress, population standard deviation, deployed thresholds $\tau_r = 10^{-3}$ and $\tau_p = 10^{-4}$, auxiliary coefficient 0.3, and $\epsilon = 10^{-6}$. Its comparison uses the same predeclared matched initialization blocks, budget, checkpoint cadence, and held-out evaluator as the baseline and DSA arms. The Fan–Liu server-run row uses the same matched blocks and the source method's exact-prefix comparison, rollout-derived transition potential, inverse-variance fusion, and action-boundary hidden states (Fan & Liu, 2026). Its interpretation remains conditional on the paper-derived hidden-state transport, terminal handling, normalization, and trainer-seam implementation used here.

D TERMINAL-REWARD BOUNDARY DETAILS

When does a terminal-reward group vary? With undiscounted binary terminal return, an all-failure group is constant and hence uninformative; a group containing both successes and failures has nonzero variance. An all-success group may be constant in the undiscounted case and can vary under discounted returns when terminal distance differs. Consequently, success contact can be necessary under one return construction without being sufficient under another. DSA is defined from the realized variance and does not assume an empirical equality between success-containing and variance-bearing groups.

Cold-start concentration of nonzero advantage. At cold start, terminal-reward batches can contain many all-failure groups. The relevant quantity for DSA is nevertheless observed group variance, not a global success-rate proxy: group size, discounting, and trajectory composition alter the mapping between the two. We therefore make no snapshot-frequency or monotonicity claim in this paper.

Dense reward as a measurability contrast. Graded rewards can produce within-group return variance before terminal success, reducing the relative need for an auxiliary progress channel. ScienceWorld tests this dense-reward boundary; its near-zero arm difference is setting-specific and does not imply that sparse reward is universal.