LOCAL EVIDENCE, JOINT ALLOCATION: PLANNING GUIDANCE FOR GOAL-CONDITIONED REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Planning can accelerate sparse-reward goal-conditioned reinforcement learning by using subgoal-conditioned behavior to regularize policy updates. Yet assigning guidance weights independently across samples couples their relative priorities with the overall balance between planning supervision and return maximization. We propose Support-Proportional Guidance (SPG), which treats guidance strength and sample allocation as a joint decision within each minibatch. A critic compares planning-induced actions with the current policy's sampled action to obtain a support count for each state-goal pair. The prevalence of sufficient support sets the mean guidance weight, while capped proportional allocation distributes the corresponding total according to support strength. The actor retains the complete planning-induced action mixture as its reference, so critic evidence changes guidance strength rather than selecting individual target actions. We establish that weak local support limits an individual guidance weight and that one changed candidate comparison has bounded influence on the entire allocation. Across six navigation and manipulation tasks, SPG improves sample efficiency over the evaluated baselines. Further results show that both excessively weak and excessively strong guidance reduce the success rate and that SPG outperforms general sample allocation methods.

1 INTRODUCTION

Goal-conditioned reinforcement learning (GCRL) learns a single policy for reaching diverse goals from shared experience (Schaul et al., 2015). Sparse rewards make distant goals difficult because useful feedback often arrives only after a long sequence of successful decisions. Hindsight experience replay and goal-conditioned supervised learning improve data reuse by treating achieved states as alternative goals (Andrychowicz et al., 2017; Ghosh et al., 2021). Planning complements these approaches with intermediate targets that connect locally achievable behaviors. Beyond selecting subgoals for execution, their associated behavior can supervise the policy conditioned on the final goal, as in RIS and PIG (Chane-Sane et al., 2021; Kim et al., 2023). Recent advances further improve subgoal reliability and execution (Han et al., 2026; Shan et al., 2026). We study the complementary question of how the resulting planning references should contribute to policy learning alongside return maximization.

The value of this supervision varies across state–goal pairs and over the course of learning. A reference that helps with one goal may be unnecessary or misleading for another, and the policy may eventually improve beyond behavior that was useful earlier. When the planner and subgoal-conditioned policy are learned online, the reference itself also evolves. Value-aware imitation addresses these differences through comparisons between reference and policy actions: Q-filter makes a binary imitation decision, whereas SPReD accounts for value uncertainty through continuous weights (Nair et al., 2018; Zhu et al., 2025). Overall guidance strength can also adapt to learning progress; TGRL adjusts teacher influence using an auxiliary reward-only learner (Shenfeld et al., 2023). Our focus is how local planning evidence should coordinate the guidance coefficients within a single minibatch update.

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.

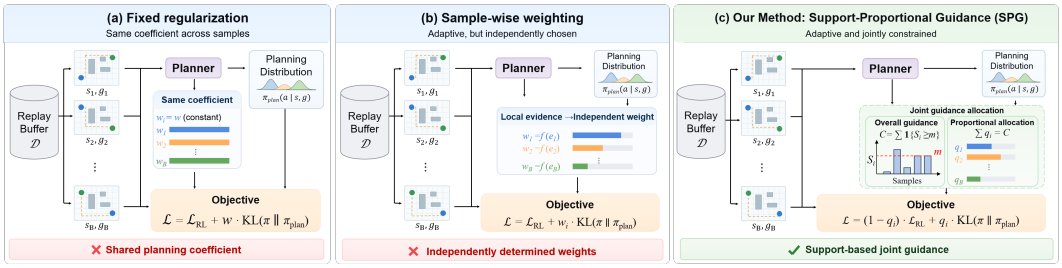


Figure 1: **From fixed regularization to joint guidance allocation.** (a) A common planning coefficient; (b) independently chosen sample weights; (c) SPG's support-derived total and proportional allocation. Vertical bars show support counts S_i against threshold m; horizontal bars show context-level loss coefficients. The planning reference π_{plan} is retained in full. The schematic objectives omit batch averaging and the common KL scale.

Figure 1 contrasts three ways of incorporating planning references into policy learning. Fixed regularization applies a common planning coefficient across replay contexts, whereas sample-wise weighting determines each coefficient from local evidence. These coefficients set both the relative priorities of individual references and their mean weight in the minibatch objective. In the independent interface, increasing one context's coefficient also increases the aggregate planning weight. These decisions need not move together: stronger evidence at a few already well-supported contexts can justify reallocating guidance without requiring a larger total. We therefore distinguish how broadly planning is supported across contexts from the relative strength of that support.

We introduce **Support-Proportional Guidance (SPG)**, which uses this distinction to determine the guidance coefficients jointly. For each state–goal pair, a critic counts sampled planning actions with higher estimated value than a sampled action from the current policy. The number of contexts meeting a support threshold determines the total guidance weight; their fraction determines its minibatch mean. This total is then shared among contexts with at least one favorable comparison, using capped proportional weights based on their support counts. Thus, contributing to the total and receiving guidance are distinct decisions. The actor combines return learning and planning supervision with complementary coefficients, retaining the complete planning-induced action mixture as each context's reference. Evidence controls the weight of that reference rather than selecting which components it contains.

The main contributions are:

- **Joint guidance allocation.** SPG coordinates the mean planning coefficient and support-dependent sample priorities within each minibatch, while retaining the complete planning reference.
- **Support-dependent allocation properties.** The analysis bounds individual weights by local support and limits the batch-wide effect of a changed candidate judgment. A conditional comparison further relates nonuniform allocation to update utility.
- **Learning efficiency and controlled evaluation.** Experiments across six navigation and manipulation tasks show improved sample efficiency. Controlled navigation studies show the value of evidence-dependent assignment and characterize the effect of guidance strength.

2 Related Work

2.1 Planning and Subgoal Learning for GCRL

Intermediate goals connect locally achievable behaviors over longer horizons. Hierarchical controllers learn subgoal selection and low-level control (Nachum et al., 2018); replay search, latent-space planning, and landmarks organize routes through intermediate states (Eysenbach et al., 2019; Nasiriany et al., 2019; Kim et al., 2021). Offline methods use latent high-level actions and entity-factored subgoals (Park et al., 2023; Haramati et al., 2026), while multistep quasimetric distances and Eikonal constraints structure temporal reachability (Zheng et al., 2026; Giammarino & Qureshi, 2026). For

reliable execution, SSE incorporates frontier experience and execution failures into graph planning, and RD-HRL addresses noisy value-based subgoal decisions offline (Han et al., 2026; Shan et al., 2026).

More directly related to SPG are methods that turn subgoal-conditioned behavior into supervision for the final-goal policy. RIS regularizes this policy toward a prior induced by imagined subgoals (Chane-Sane et al., 2021). PIG uses graph paths for self-imitation and also introduces subgoal skipping during execution (Kim et al., 2023). In offline GCRL, Subgoal Advantage-Weighted Policy Bootstrapping (SAW) uses advantage-weighted importance sampling over in-trajectory subgoals to train a flat policy without learning a subgoal generator (Zhou & Kao, 2025). Their weighting acts on the behavior used to supervise the final-goal policy. In SPG, a complete planning-induced mixture remains attached to each replay context. The allocation coordinates the reference-loss coefficients across those contexts, rather than choosing the components that enter each reference.

2.2 Adaptive Guidance and Policy Regularization

The usefulness of a reference depends on both its value and its compatibility with the learner. Demonstration-Regularized RL analyzes how expert demonstrations reduce sample complexity through KL regularization toward a behavior-cloned policy (Tiapkin et al., 2024). However, the choice of behavioral reference policy class can itself lead to slow, unstable, or suboptimal learning (Rudner et al., 2021). Adaptive imitation responds to different sources of mismatch. Q-filter admits demonstration losses through critic comparisons (Nair et al., 2018), whereas SPReD uses ensembles to model uncertainty in demonstration and policy action values and construct continuous regularization weights (Zhu et al., 2025). ADVISOR addresses the teacher–student information gap by adapting imitation to what the student can reproduce (Weihs et al., 2021).

Overall guidance can also adapt. TGRL uses a reward-only auxiliary learner as a performance reference for adjusting teacher influence (Shenfeld et al., 2023). CADENT combines automaton and policy knowledge through experience-gated state–action guidance in source-to-target transfer (Alinejad et al., 2026). Other interfaces select teachers, propose imitation-based actions, or use guide-policy exploration curricula (Kurenkov et al., 2020; Hu et al., 2024; Uchendu et al., 2023). These approaches use different evidence sources and control variables. SPG couples the coefficients of one minibatch through a support-derived total and individual caps, with favorable counts setting relative priorities.

2.3 Constrained Optimization and Allocation

Weighted proportional fairness allocates shared resources through a weighted logarithmic objective (Kelly et al., 1998). SPG uses this criterion for planning-loss coefficients, constructing the total and priorities from planning evidence. Information-projection theory offers another view of constrained allocation (Csiszár, 1975); the equivalence under the fixed-total and individual-cap constraints is derived in Appendix A.1.

Other optimization methods operate on different objects. Relative-entropy policy search, maximum a posteriori policy optimization, and regularized MDP theory constrain or regularize policy distributions (Peters et al., 2010; Abdolmaleki et al., 2018; Geist et al., 2019). PCGrad and CAGrad coordinate task gradients through conflict projection or worst-task local improvement (Yu et al., 2020; Liu et al., 2021); Bloop optimizes an auxiliary objective over the main objective's minimizers (Hsieh et al., 2024). Prioritized replay changes transition sampling (Schaul et al., 2016). SPG retains distributional KL regularization but constrains the loss coefficients within the sampled minibatch, leaving replay-sampling and optimizer procedures unchanged.

3 Preliminaries

3.1 Goal-Conditioned Reinforcement Learning

A goal-conditioned MDP consists of states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, goals $g \in \mathcal{G}$, dynamics P, reward $r(s, a, g)$, and discount $\gamma \in [0, 1)$. The policy $\pi_\theta(a \mid s, g)$ maximizes expected discounted return over goals drawn from p_G. We use an off-policy actor–critic with hindsight relabeling (Schaul et al.,

2015; Andrychowicz et al., 2017). For a replay context $x_i = (s_i, g_i)$, policy improvement uses the return loss $\mathcal{L}_{\mathrm{RL},i}(\theta) = -\mathbb{E}_{a\sim\pi_\theta(\cdot|x_i)} Q_\phi(s_i, a, g_i)$, where $Q_\phi = \min_h Q_{\phi,h}$ is the online twin-critic estimate (Fujimoto et al., 2018).

3.2 Distributional Planning Guidance

The planner provides K intermediate targets $z_{i,1:K}$ for each context. Applying the target actor π_T to these subgoals defines the planning reference

$$\pi_{\mathrm{plan}}(a \mid x_i) = \frac{1}{K} \sum_{k=1}^{K} \pi_T(a \mid s_i, z_{ik}). \tag{1}$$

The dependence on the sampled subgoals is implicit. Thus, π_{plan} is a mixture of action densities, not an average action or an additional learned policy. The learned-subgoal case follows RIS's policy-prior formulation; graph paths can also supply intermediate targets to this distributional interface (Chane-Sane et al., 2021; Kim et al., 2023). The planning loss is

$$\mathcal{L}_{\mathrm{P},i}(\theta) = \alpha D_{\mathrm{KL}}(\pi_\theta(\cdot \mid x_i) \,\|\, \pi_{\mathrm{plan}}(\cdot \mid x_i)), \qquad \alpha > 0.$$

Reference parameters are fixed during actor differentiation; the planner, critic, and target networks follow the backbone update rules. Conventional additive regularization averages $\mathcal{L}_{\mathrm{RL},i} + \mathcal{L}_{\mathrm{P},i}$ over the batch.

4 Support-Proportional Guidance

SPG computes a vector of planning coefficients for each replay minibatch, coupling their total while retaining support-dependent differences between contexts.

4.1 Planning Evidence and Total Guidance Weight

For each of B replay contexts, draw one actor action $a_i \sim \pi_\theta(\cdot \mid x_i)$ and one evidence action $\widetilde{a}_{ik} \sim \pi_T(\cdot \mid s_i, z_{ik})$ per planning component. The same K subgoals define these components and Eq. equation 1. All comparisons use the final goal g_i and the target twin-critic minimum $\bar{Q} = \min_h Q_h^-$:

$$S_i = \sum_{k=1}^{K} \mathbf{1}\{\bar{Q}(s_i, \widetilde{a}_{ik}, g_i) > \bar{Q}(s_i, a_i, g_i)\}. \tag{2}$$

Set $m = \lceil \eta K \rceil$ for $\eta \in (0, 1]$ and $C = \sum_{i=1}^{B} \mathbf{1}\{S_i \geq m\}$. We use C/B, the fraction of sufficiently supported contexts, as the mean planning coefficient. After a context passes, further favorable comparisons can change its share but not its contribution to C. Counts retain frequency rather than value-gap magnitude and are invariant to positive affine transformations of the gate critic at fixed actions.

Guidance is allocated over $E = \{i : S_i > 0\}$, with $n = |E|$. Contexts with $0 < S_i < m$ can therefore receive partial supervision without contributing to C. Every passing context is eligible, giving $0 \leq C \leq n$ and ensuring feasibility under the individual caps.

4.2 Joint Proportional Allocation

At the prescribed total C, SPG allocates larger coefficients to contexts with stronger support. For $0 < C < n$, the weights satisfy

$$q_i = \min\{1, \lambda S_i\} \quad (i \in E), \qquad \sum_{i=1}^{B} q_i = C, \tag{3}$$

Outside E, $q_i = 0$. The total uniquely determines $\lambda > 0$ when $0 < C < n$. The boundary cases are $q = 0$ for $C = 0$ and $q = \mathbf{1}_E$ for $C = n$. The cap keeps both complementary actor coefficients nonnegative.

Algorithm 1 Support-Proportional Guidance: one active actor update

Require: Replay batch $x_{1:B}$; actor, planner, target actor and critics; K, η
1: Set $m = \lceil \eta K \rceil$. Obtain K subgoals per context and form π_{plan} using Eq. equation 1.
2: Draw one actor action a_i per context and one evidence action $\widetilde{a}_{ik}$ from each of its K planning components.
3: Compute S_i by Eq. equation 2; set $C = \sum_i \mathbf{1}\{S_i \geq m\}$, $E = \{i : S_i > 0\}$, and $n = |E|$.
4: **if** $C = 0$ **then**
5: Set every $q_i = 0$.
6: **else if** $C = n$ **then**
7: Set $q_i = \mathbf{1}\{i \in E\}$.
8: **else**
9: Solve $\sum_{i \in E} \min(1, \lambda S_i) = C$ and set q_i by Eq. equation 3.
10: **end if**
11: Evaluate $\widehat{\mathcal{L}}_{\text{RL},i}$ and $\widehat{\mathcal{L}}_{\text{P},i}$ using the same actor draw and the full planning reference.
12: Hold q and reference parameters fixed; retain gradients through the actor action.
13: Take one backbone actor-optimizer step on the sampled form of Eq. equation 4.

Unsaturated weights preserve support-count ratios. When a weight reaches one, the remaining mass is redistributed among the unsaturated contexts. For $0 < C < n$, this rule maximizes $\sum_{i \in E} S_i \log q_i$ under the total and cap constraints, giving a support-weighted proportional-fair allocation (Kelly et al., 1998). The logarithm reduces the marginal allocation utility of further concentrating weight. The characterization and solver are given in Appendix A.1.

4.3 Joint Actor Update

For the computed allocation q, the actor objective is

$$\mathcal{L}_{\text{SPG}}(\theta; q) = \frac{1}{B} \sum_{i=1}^{B} \left[(1 - q_i) \mathcal{L}_{\text{RL},i}(\theta) + q_i \mathcal{L}_{\text{P},i}(\theta) \right]. \tag{4}$$

Planning supervision replaces part of each context's return objective, with mean coefficients C/B and $1 - C/B$, respectively. The scale α is included in $\mathcal{L}_{\text{P},i}$. These weights control the loss composition; the parameter change also depends on its gradients and the optimizer.

The implemented update substitutes sampled losses $\widehat{\mathcal{L}}_{\text{RL},i}$ and $\widehat{\mathcal{L}}_{\text{P},i}$ in Eq. equation 4. One reparameterized actor draw is shared by the return term, KL log-density ratio, and support comparisons. The realized allocation and reference parameters are fixed during differentiation; gradients through the actor action are retained in the online critic and both log densities. The resulting step differentiates this sampled actor surrogate with the observed allocation held fixed.

The KL is evaluated against the complete mixture π_{plan}, rather than averaged over component-wise KL losses. All reference components therefore remain in the planning loss. Algorithm 1 summarizes the update; numerical and training details are in Appendix B.

4.4 Properties of the Guidance Allocation

The following result describes weakly supported contexts and propagation of a changed judgment. Let $q = A_m(S)$ be the complete count-to-coefficient map, including the total $C(S)$.

Theorem 1 (Support envelope and bounded influence)**.** *Fix integers $K \geq 1$ and $1 \leq m \leq K$. For $S \in \{0, \ldots, K\}^B$ and $q = A_m(S)$, every weight satisfies $q_i \leq \min(1, S_i/m)$. If $S_j < K$, let $S' = S + \mathbf{e}_j$, $q' = A_m(S')$, $C = C(S)$, and $C' = C(S')$. Then*

$$\|q' - q\|_1 \leq \begin{cases} 2/m, & C' = C, \\ 1, & C' = C + 1. \end{cases} \tag{5}$$

When $C' = C + 1$, every weight is nondecreasing and the distance equals one. Reversing a judgment gives the reversed comparison.



(a) (b) (c) (d) (e) (f)

Figure 2: Visualizations of our experimental environments: (a) AntMaze-U, (b) AntMaze-S, (c) AntMaze-Π, (d) AntMaze-W, (e) FetchPush, (f) FetchPickAndPlace.

For $0 < S_i < m$, the envelope gives $q_i < 1$, retaining a nonzero coefficient on $\mathcal{L}_{\mathrm{RL},i}$. At scale $1/m$, every passing context contributes one unit, so $\sum_i \min(1, S_i/m) \geq C$; the required scaling cannot be larger. The same construction limits how a local judgment changes the whole allocation. At fixed total, Eq. equation 5 bounds all redistributed weight. Crossing the threshold adds one unit to the total without decreasing any context's weight. Thus weak local support remains limited even when support elsewhere is strong. The proof and fixed-gradient extension are in Appendix A.2.

For the learning comparison, fix the replay contexts, network parameters, references, actor samples, and optimizer state, and compare q with $b = (C/n)\mathbf{1}_E$ for $n > 0$. Only the coefficients differ between these two updates. For a differentiable evaluation objective J, assume its composition with the optimizer step has a locally Lipschitz gradient along the comparison segment. If the first-order benefit of planning relative to return learning has a positive linear dependence on S_i/K up to a bounded residual, Proposition A.2 bounds the change in J. The support-associated gain must exceed the residual penalty and finite-step curvature term; when it does, SPG improves J over the uniform update. Appendix A.3 gives the precise bound and proof.

5 Experiments

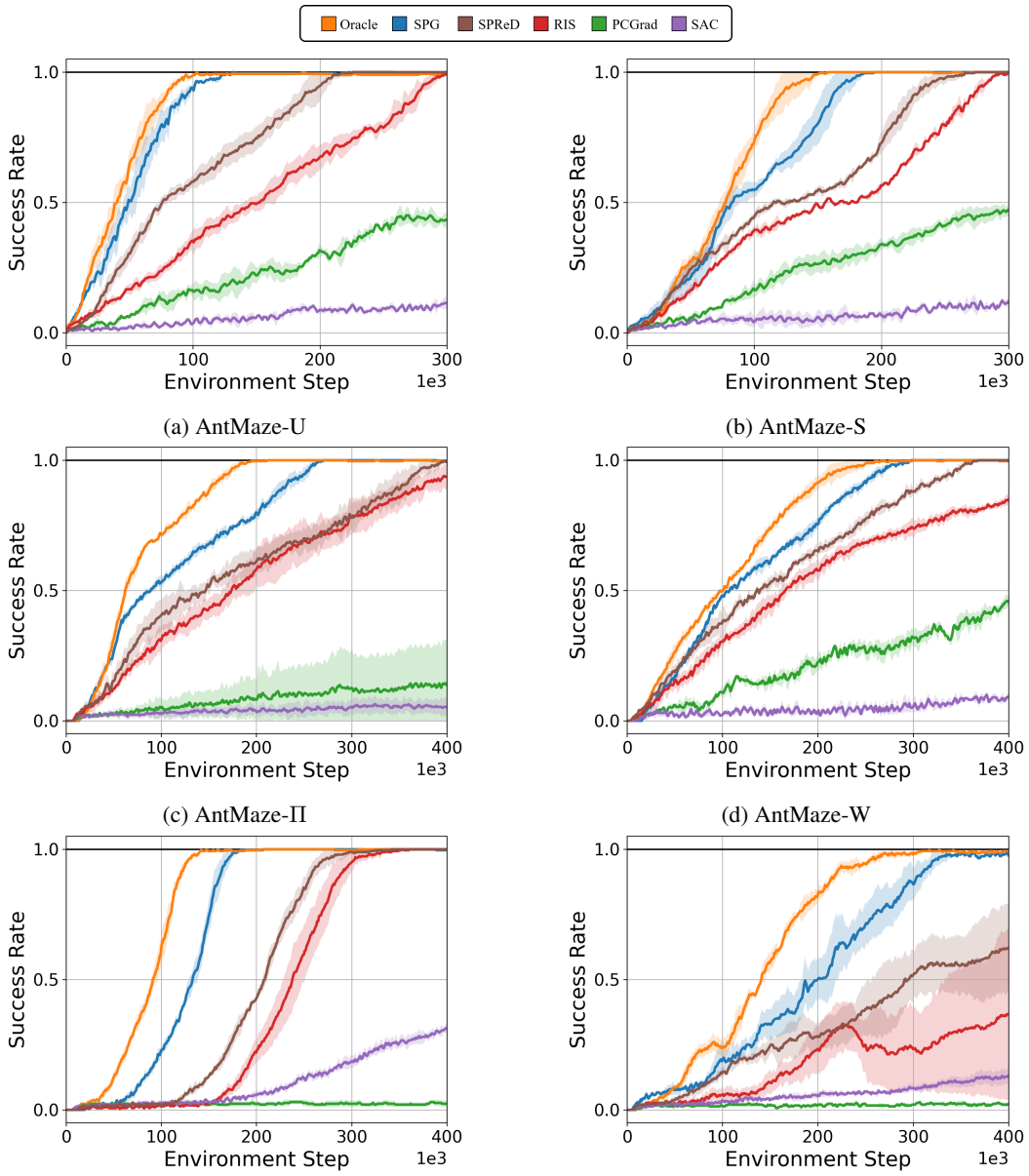
We evaluate whether SPG makes more effective use of a given planning source, then examine the guidance coefficients and the evidence used to construct them.

5.1 Experimental Setup

Tasks and planning sources. The main study uses the RIS planner on AntMaze-U, S, Π, and W and on FetchPush and FetchPickAndPlace (Figure 2) (Chane-Sane et al., 2021; Plappert et al., 2018). Budgets are 300K environment interactions on U and S and 400K otherwise. A separate comparison retains PIG's graph planner on AntMaze-U and W, with 300K and 3M interactions, respectively (Kim et al., 2023). Comparisons therefore share a planning source and budget within each task, rather than matching budgets across planning sources.

Implementation. All methods use an off-policy goal-conditioned actor–critic framework with HER. The learned-subgoal backbone uses twin critics and the clipped estimate $Q_\phi = \min(Q_{\phi,1}, Q_{\phi,2})$ to mitigate overestimation (Fujimoto et al., 2018). SPG uses the target estimate $\bar{Q} = \min(Q_1^-, Q_2^-)$ for support evaluation and the online estimate for the return loss. Actor and critic targets use Polyak averaging with coefficient 5×10^{-3}. The regularized comparisons share the full-mixture reference construction in Eq. equation 1. SPG detaches its continuous coefficients during actor differentiation while retaining action-path gradients. The learned-subgoal settings are $K = 16$, $\eta = 0.25$, $\alpha = 0.1$, and $B = 2048$; initialization and further hyperparameters are in Appendix B.

Baselines. **SAC** provides the unguided actor–critic comparison with HER, learning without a planning regularizer (Haarnoja et al., 2018). **RIS** uses fixed-KL regularization: every replay context contributes $\mathcal{L}_{\mathrm{RL},i} + \mathcal{L}_{\mathrm{P},i}$, with the common KL scale already included in $\mathcal{L}_{\mathrm{P},i}$ (Chane-Sane et al., 2021). **SPReD** was developed for learning with previously collected demonstrations. We adapt its uncertainty-aware smooth weighting to online GCRL using the evolving planning reference π_{plan} and online replay batches (Zhu et al., 2025). It tests local evidence-based weighting rather than a shared coefficient constraint. **PCGrad** treats the return and planning losses as two objectives and projects



(a) AntMaze-U (b) AntMaze-S

(c) AntMaze-Π (d) AntMaze-W

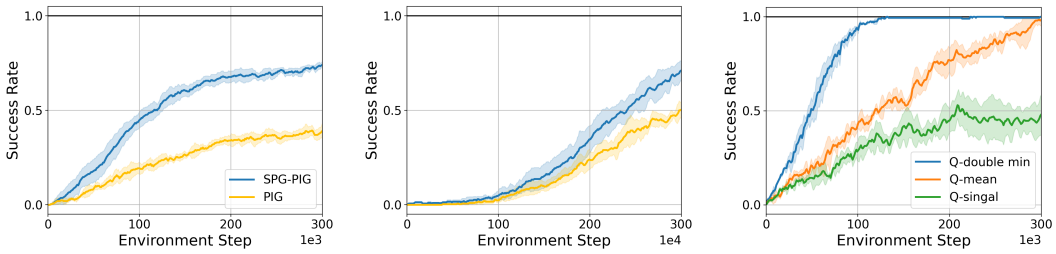
(e) FetchPush (f) FetchPickAndPlace

Figure 3: **Learning with the RIS planning source.** Random-goal success versus environment interactions. Curves show means with standard-deviation shading over four seeds. Oracle replaces only SPG's support-evaluation critic; the planner, reference construction, and allocation rule are unchanged.

conflicting components of their actor gradients before combining the updates (Yu et al., 2020). Our code is available at `https://anonymous.4open.science/r/SPG-7D0B/`.

Oracle substitutes an oracle critic for support evaluation while retaining SPG's continuous allocation, planning reference, and online return critic. This privileged variant assesses attainable learning performance under near-optimal support estimates within the same SPG framework.

Evaluation. Learning curves report success on random start–goal pairs, averaged over four seeds with standard-deviation shading. The fixed-goal guidance controls in Figure 5 report three seeds and sample standard deviations. The η table summarizes within-study success-curve AUC over four



(a) AntMaze-U: 300K interactions (b) AntMaze-W: 3M interactions (c) AntMaze-U: critic configuration

Figure 4: **Planning-source and critic comparisons.** (a)–(b) PIG and SPG-PIG with the same graph-planning procedure, evaluated for 300K interactions on AntMaze-U and 3M on AntMaze-W. (c) Minimum, mean, and single-critic configurations in SPG on AntMaze-U. All curves show means with standard-deviation shading over four seeds.

seeds, weighting the two layouts equally. Additional random- and farthest-goal results follow the separate paired-study protocol in Appendix C.2.

5.2 Main Results

Learning efficiency and final success. SPG reaches high success earlier across the six tasks in Figure 3. On AntMaze-U, AntMaze-S, and FetchPush, several guided methods eventually approach full success; the distinction is how soon they acquire that behavior. On AntMaze-Π and AntMaze-W, SPG maintains a lead over much of training. FetchPickAndPlace exhibits both earlier improvement and a substantial terminal gap to RIS and SPReD. The benefit is therefore not confined to either faster learning or the last checkpoint: its form depends on how much the other methods learn within the available budget.

SAC remains low despite hindsight relabeling, while RIS and SPReD benefit from planning. SPG improves on both reference-using baselines; its gains concern the use of an existing planning source. PCGrad also trails these methods. The assignment controls below retain the complementary loss form to distinguish evidence-based allocation from differences between complete training interfaces.

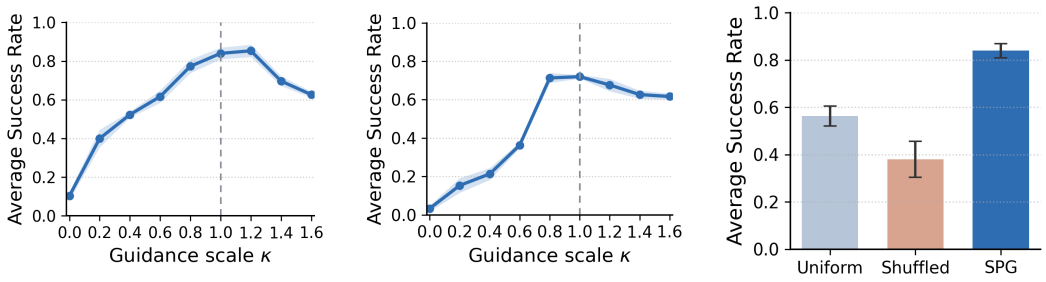
Performance with oracle support estimates. Oracle often reaches high success sooner, especially on AntMaze-Π and FetchPickAndPlace, although its terminal performance is close to learned-critic SPG on several tasks. The remaining gap is therefore partly a delay in learning behavior that SPG eventually acquires. Better support evaluation offers a route to reduce that delay without replacing the planner or selecting a narrower reference.

Graph-derived references. SPG-PIG achieves higher success than PIG in both layouts (Figure 4(a)–(b)). The separation persists through 300K interactions on AntMaze-U and the later stages of the 3M AntMaze-W run. Each pair retains its graph planner. The gains thus extend beyond RIS-derived subgoals to the policy-learning interface of a different planning source.

5.3 Guidance Strength and Evidence-Based Allocation

Guidance strength. Scaling the support-derived total by κ gives a non-monotonic response in both layouts (Figure 5(a)–(b)). AntMaze-U success rises from 0.103 without guidance to 0.840 at $\kappa = 1$, then falls to 0.627 at 1.6; the peak is 0.853 at 1.2. AntMaze-W peaks at the default with 0.720. The original rule provides a common effective setting despite different peak locations. In Eq. equation 4, increasing planning coefficients also decreases return coefficients. More guidance can therefore displace useful return-based learning rather than simply adding supervision.

Evidence-dependent assignment. On AntMaze-U, *Uniform* uses $q_i = C/n$ within E, while *Shuffled* permutes SPG's coefficients within E. All rules retain the full reference and complementary objective. Uniform treats weak and strong support identically, whereas SPG retains their relative



(a) AntMaze-U: guidance scale (b) AntMaze-W: guidance scale (c) AntMaze-U: allocation rule

Figure 5: **Guidance strength and evidence-based allocation.** (a)–(b) Fixed-goal success under multiplier κ; dashed lines mark the original rule, $\kappa = 1$. (c) Uniform, shuffled, and support-proportional assignments on AntMaze-U at the original guidance total. Means are over three seeds; shading and error bars show sample standard deviations.

Table 1: **Support-fraction sensitivity.** Mean success-curve AUC over four seeds, with an equal-layout average. Bold marks the default.

η	AntMaze-U AUC	AntMaze-W AUC	Layout-equal mean
0.125	0.401	0.228	0.315
0.25	**0.429**	**0.264**	**0.346**
0.375	0.417	0.247	0.332
0.50	0.386	0.221	0.304

priorities. Shuffling preserves the coefficient multiset of a given batch, including its total and concentration. SPG attains 0.840 success, versus 0.563 for Uniform and 0.380 for Shuffled (Figure 5(c)). Both the relative shares and their correspondence with support matter: unequal weights alone are insufficient to explain the gain. The rules are recomputed on their respective training batches.

5.4 Evidence Estimation and the Support Threshold

Figure 4(c) examines the clipped estimate used in SPG. The twin-critic minimum reaches near-full success earlier than the mean, while the single critic remains much lower. Because value comparisons determine both threshold membership and relative shares, the evidence critic affects several coefficients at once. This favors the clipped configuration and complements Oracle's evidence-quality comparison.

At $K = 16$, Table 1 tests $m = 2, 4, 6, 8$. The default $\eta = 0.25$ has the highest reported AUC in both layouts; 0.375 is close. Lower and higher thresholds perform worse. Raising m can reduce the admitted total while leaving eligibility unchanged, unlike rescaling with κ. The ordering favors an intermediate support requirement over the evaluated AUC window.

6 Conclusion

SPG coordinates planning supervision through a support-derived total and capped proportional shares. Its allocation bounds the influence of finite evidence while preserving complete planning references. Results with learned and graph-based planners show improved goal-reaching performance, and controlled studies reveal the value of matching guidance coefficients to local support.

Limitations remain in evidence quality and task coverage. Support counts omit comparison magnitudes, and bounded allocation sensitivity does not correct systematic critic bias. The Oracle gap motivates uncertainty-aware support estimation; adaptive candidate sampling could concentrate computation where evidence is most ambiguous. The evaluation is also limited to state-based simulated tasks with explicitly evaluable reference mixtures. Extending SPG to visual observations and more expensive planners requires assessing both the reliability and cost of these references.

AI USE STATEMENT

Generative AI tools assisted literature review, manuscript preparation, theoretical proofs. The authors take responsibility for the final text, mathematical claims, citations, and experimental record.

REPRODUCIBILITY STATEMENT

Algorithm 1 and Appendix B specify the actor update, loss estimator, optimizer state, initialization, and allocation solver. Full proofs are in Appendix A; experimental protocols and reporting details are in Appendix C. Supplementary data provide per-seed curves, coefficient diagnostics, and fixed-state comparisons. The associated scripts compute the reported summaries; the available data and computational scope are specified in Appendix C.

REFERENCES

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=S1ANxQW0b.

Mahyar Alinejad, Yue Wang, and George K. Atia. CADENT: Gated hybrid distillation for sample-efficient transfer in reinforcement learning. In *Proceedings of The 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 4051–4059. PMLR, 2026. URL https://proceedings.mlr.press/v300/alinejad26a.html.

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*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/453fadbd8a1a3af50a9df4df899537b5-Abstract.html.

Elliot Chane-Sane, Cordelia Schmid, and Ivan Laptev. Goal-conditioned reinforcement learning with imagined subgoals. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1430–1440. PMLR, 2021. URL https://proceedings.mlr.press/v139/chane-sane21a.html.

Imre Csiszár. I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, 3(1):146–158, 1975. doi: 10.1214/aop/1176996454. URL https://web.mit.edu/6.291/www/handouts/csiszar_Idiv.pdf.

Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. Search on the replay buffer: Bridging planning and reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/5c48ff18e0a47baaf81d8b8ea51eec92-Abstract.html.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1587–1596. PMLR, 2018. URL https://proceedings.mlr.press/v80/fujimoto18a.html.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized Markov decision processes. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2160–2169. PMLR, 2019. URL https://proceedings.mlr.press/v97/geist19a.html.

Dibya Ghosh, Abhishek Gupta, Ashwin Reddy, Justin Fu, Coline Devin, Benjamin Eysenbach, and Sergey Levine. Learning to reach goals via iterated supervised learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=rALA0Xo6yNJ.

Vittorio Giammarino and Ahmed Hussain Qureshi. Goal reaching with eikonal-constrained hierarchical quasimetric reinforcement learning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/60d166781779c460d8ad83d78f2ba03a-Abstract-Conference.html.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Seungyul Han, Jaebak Hwang, Sanghyeon Lee, and Jeongmo Kim. Strict subgoal execution: Reliable long-horizon planning in hierarchical reinforcement learning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/ec94195bca07a55e83968fbfa6efb8b0-Abstract-Conference.html.

Dan Haramati, Carl Qi, Tal Daniel, Amy Zhang, Aviv Tamar, and George D. Konidaris. Hierarchical entity-centric reinforcement learning with factored subgoal diffusion. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/de2bc3eed995c24307e8011161ac1438-Abstract-Conference.html.

Yu-Guan Hsieh, James Thornton, Eugene Ndiaye, Michal Klein, Marco Cuturi, and Pierre Ablin. Careful with that scalpel: Improving gradient surgery with an EMA. In *International Conference on Machine Learning*, volume 235, pp. 19085–19100, 2024. URL https://proceedings.mlr.press/v235/hsieh24a.html.

Hengyuan Hu, Suvir Mirchandani, and Dorsa Sadigh. Imitation bootstrapped reinforcement learning. In *Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.056. URL https://www.roboticsproceedings.org/rss20/p056.html.

F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate control for communication networks: Shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49(3): 237–252, 1998. doi: 10.1057/palgrave.jors.2600523. URL https://link.springer.com/article/10.1057/palgrave.jors.2600523.

Junsu Kim, Younggyo Seo, and Jinwoo Shin. Landmark-guided subgoal generation in hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/ee39e503b6bedf0c98c388b7e8589aca-Abstract.html.

Junsu Kim, Younggyo Seo, Sungsoo Ahn, Kyunghwan Son, and Jinwoo Shin. Imitating graph-based planning with goal-conditioned policies. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6lUEy1J5R7p.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Andrey Kurenkov, Ajay Mandlekar, Roberto Martin-Martin, Silvio Savarese, and Animesh Garg. AC-Teach: A bayesian actor-critic method for policy learning with an ensemble of suboptimal teachers. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 717–734. PMLR, 2020. URL https://proceedings.mlr.press/v100/kurenkov20a.html.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/9d27fdf2477ffbff837d73ef7ae23db9-Abstract.html.

Ofir Nachum, Shixiang Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/e6384711491713d29bc63fc5eeb5ba4f-Abstract.html.

Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE International Conference on Robotics and Automation*, pp. 6292–6299. IEEE, 2018. doi: 10.1109/ICRA.2018.8463162.

Soroush Nasiriany, Vitchyr Pong, Steven Lin, and Sergey Levine. Planning with goal-conditioned policies. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/c8cc6e90ccbff44c9cee23611711cdc4-Abstract.html.

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. HIQL: Offline goal-conditioned RL with latent states as actions. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/6d7c4a0727e089ed6cdd3151cbe8d8ba-Abstract-Conference.html.

Jan Peters, Katharina Mülling, and Yasemin Altun. Relative entropy policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pp. 1607–1612, 2010. URL https://ojs.aaai.org/index.php/AAAI/article/view/7727.

Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, Vikash Kumar, and Wojciech Zaremba. Multi-goal reinforcement learning: Challenging robotics environments and request for research, 2018. URL https://arxiv.org/abs/1802.09464. arXiv preprint arXiv:1802.09464.

Tim G. J. Rudner, Cong Lu, Michael A. Osborne, Yarin Gal, and Yee Whye Teh. On pathologies in KL-regularized reinforcement learning from expert demonstrations. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/eecca5b6365d9607ee5a9d336962c534-Abstract.html.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1312–1320. PMLR, 2015. URL https://proceedings.mlr.press/v37/schaul15.html.

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.

Yixiang Shan, Haipeng Liu, Ting Long, and Yi Chang. RD-HRL: Generating reliable sub-goals for long-horizon sparse-reward tasks. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/56d5bf1977917d00c6e4dd9cf8401741-Abstract-Conference.html.

Idan Shenfeld, Zhang-Wei Hong, Aviv Tamar, and Pulkit Agrawal. TGRL: An algorithm for teacher guided reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31077–31093. PMLR, 2023. URL https://proceedings.mlr.press/v202/shenfeld23a.html.

Daniil Tiapkin, Denis Belomestny, Daniele Calandriello, Eric Moulines, Aleksei Naumov, Pierre Perrault, Michal Valko, and Pierre Ménard. Demonstration-regularized RL. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/7b5d3047939b63ed97fcbbee23f8eb77-Abstract-Conference.html.

Ikechukwu Uchendu, Ted Xiao, Yao Lu, Banghua Zhu, Mengyuan Yan, Joséphine Simon, Matthew Bennice, Chuyuan Fu, Cong Ma, Jiantao Jiao, Sergey Levine, and Karol Hausman. Jump-start

reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 34556–34583. PMLR, 2023. URL https://proceedings.mlr.press/v202/uchendu23a.html.

Luca Weihs, Unnat Jain, Iou-Jen Liu, Jordi Salvador, Svetlana Lazebnik, Aniruddha Kembhavi, and Alexander Schwing. Bridging the imitation gap by adaptive insubordination. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/file/9fc664916bce863561527f06a96f5ff3-Paper.pdf.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3fe78a8acf5fda99de95303940a2420c-Abstract.html.

Bill Zheng, Vivek Myers, Benjamin Eysenbach, and Sergey Levine. Scaling goal-conditioned reinforcement learning with multistep quasimetric distances. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/eddc0fab5a42f8d8de6eb5566cd9f1d3-Abstract-Conference.html.

John L. Zhou and Jonathan C. Kao. Flattening hierarchies with policy bootstrapping. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/77af217d857e3a84e3249c727d903477-Abstract-Conference.html.

Yujie Zhu, Charles A. Hepburn, Matthew Thorpe, and Giovanni Montana. Uncertainty-based smooth policy regularisation for reinforcement learning with few demonstrations. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/7554259f1a08e2c679398722669c63c7-Abstract-Conference.html.

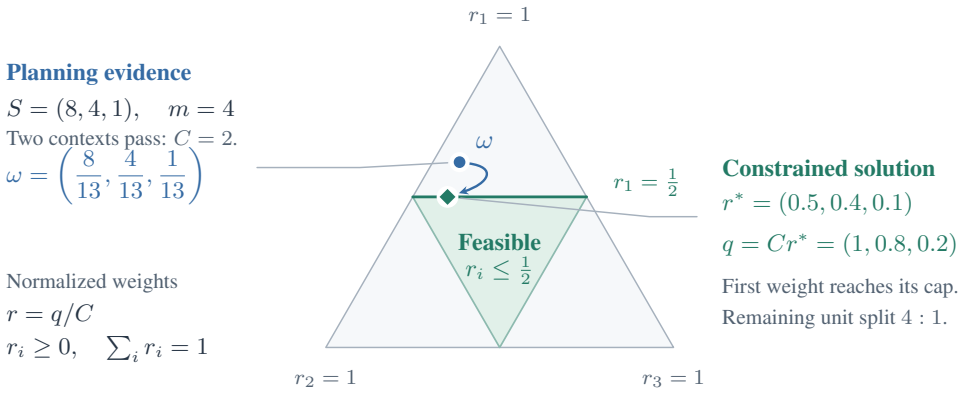


Figure 6: **Capped proportional allocation.** The shaded region enforces $r_i \leq 1/2$ for $r = q/C$, $C = 2$. The evidence proportions ω map to their constrained KL minimizer r^*, yielding $q = (1, 0.8, 0.2)$. The arrow is schematic.

A Proofs of the Allocation and Update Properties

All update comparisons below fix the replay contexts, network parameters, planning references, and actor samples for one active update. Support counts and weights are held constant during differentiation. The allocation results cover every support-count vector; the comparison in Proposition A.2 also requires a relation between support and update utility. The proofs use exact arithmetic, with implementation tolerances specified in Appendix B.

A.1 Capped proportional fairness and information projection

For completeness, the allocation in Eq. equation 3 can be characterized by the following constrained objective:

$$\max_q \sum_{i \in E} S_i \log q_i \quad \text{subject to} \quad 0 \leq q_i \leq 1, \quad \sum_{i \in E} q_i \leq C, \qquad q_i = 0 \ (i \notin E). \tag{A.1}$$

Proposition A.1 (Capped proportional allocation)**.** *For $0 < C < n$, Eq. equation A.1 has the unique solution*

$$q_i = \min\{1, \lambda S_i\} \ (i \in E), \qquad \sum_i q_i = C.$$

with a unique positive λ. For $C = 0$, set $q = 0$; for $C = n$, set $q = \mathbf{1}_E$. For $C > 0$, define $\omega_i = S_i / \sum_{j \in E} S_j$ on E. The same allocation uniquely minimizes $D_{\mathrm{KL}}(q_E/C \,\|\, \omega)$ over $q_E \in [0,1]^E$ with $\sum_{i \in E} q_i = C$.

Figure 6 illustrates why enforcing an individual cap requires redistributing the remaining weight rather than simply clipping normalized shares.

Let E have $n > 0$ coordinates with scores $s_i > 0$. For a prescribed total $c \in (0, n]$, define

$$\mathcal{Q}_c = \{q \in [0,1]^E : \mathbf{1}^\top q = c\}, \qquad f_s(\lambda) = \sum_{i \in E} \min(1, \lambda s_i).$$

The proportional-fair objective is optimized over $\mathcal{Q}_{\leq c} = \{q \in [0,1]^E : \mathbf{1}^\top q \leq c\}$, whereas the information projection uses $\mathcal{Q}_c$. We allow real-valued c in this derivation; SPG uses the integer-valued choice $C(S)$.

Proof of Proposition A.1. First suppose $0 < c < n$. Extend $\sum_i s_i \log q_i$ to take the value $-\infty$ whenever an eligible coordinate is zero. The extended objective is upper semicontinuous on the compact set $\mathcal{Q}_{\leq c}$, which contains the strictly positive vector $(c/n)\mathbf{1}$. A maximizer therefore exists and has finite objective value, so all its eligible coordinates are positive. Strict concavity on this domain makes the maximizer unique. Its total must equal c: otherwise, at least one coordinate is below one and can be increased to improve the objective.

Slater's condition for the inequality problem holds at $(c/(2n))\mathbf{1}$. With multiplier $\xi \geq 0$ for the total and $\tau_i \geq 0$ for $q_i \leq 1$, the maximization Lagrangian is

$$\mathcal{L}(q, \xi, \tau) = \sum_i s_i \log q_i - \xi\Big(\sum_i q_i - c\Big) - \sum_i \tau_i(q_i - 1).$$

Stationarity and complementary slackness give

$$\frac{s_i}{q_i} = \xi + \tau_i, \qquad \tau_i(q_i - 1) = 0.$$

Because $c < n$, at least one coordinate is unsaturated, which implies $\xi > 0$. Unsaturated coordinates satisfy $q_i = s_i/\xi$; saturated coordinates satisfy $s_i \geq \xi$. Hence $q_i = \min(1, \lambda s_i)$ with $\lambda = 1/\xi$. The function f_s is continuous, starts at zero, reaches n, and is strictly increasing below n. Thus $f_s(\lambda) = c$ has exactly one positive root. When $c = n$, the only vector with this total is $\mathbf{1}$, which also uniquely maximizes the objective under the inequality constraint. The multiplier producing this vector need not be unique. When $c = 0$, we define the allocation to be zero directly, without using a logarithmic objective or normalizing the weights.

For the information-projection statement, take $c > 0$, put $T = \sum_i s_i$, $\omega_i = s_i/T$, and $r = q/c$. Minimizing

$$D_{\mathrm{KL}}(r\|\omega) = \sum_i r_i \log(r_i/\omega_i) \quad \text{over} \quad \sum_i r_i = 1, \quad 0 \leq r_i \leq 1/c$$

is a strictly convex problem. If $c < n$, every coordinate of its minimizer is positive. Transferring an arbitrarily small amount from a positive coordinate to a zero coordinate would otherwise decrease the objective, since the one-sided derivative of $r \log r$ at zero is $-\infty$. The KKT conditions therefore give

$$r_i = \min(1/c, a\omega_i)$$

for a common positive a. Multiplying by c recovers the capped proportional form and thus the same unique solution with the prescribed total. If $c = n$, the feasible set contains a single point. This completes the proof. □

Proportional fairness and information projection give the same allocation through different objectives (Kelly et al., 1998; Csiszár, 1975). First-order optimality also gives, for any $w \in \mathcal{Q}_c$ with $c > 0$,

$$\sum_{i \in E} s_i \frac{w_i - q_i}{q_i} \leq 0.$$

This is the weighted proportional-fairness condition at total c.

Exact sorted solver. Order the positive scores as $s_{(1)} \geq \cdots \geq s_{(n)}$ and define $T_r = \sum_{j=r+1}^{n} s_{(j)}$ for $0 \leq r < n$, with $T_n = 0$. Since the saturated coordinates form a prefix,

$$f_s(\lambda) = \min_{0 \leq r \leq n} \{r + \lambda T_r\}.$$

For $0 < c < n$, the $r = n$ branch is above c. Set

$$\lambda^* = \max_{0 \leq r < n} \frac{c - r}{T_r}. \tag{A.2}$$

At λ^*, every affine branch is at least c and one equals c, so $f_s(\lambda^*) = c$. All denominators are positive tail sums. Sorting and computing these sums therefore gives a direct alternative to bisection, with no denominator offset.

A.2 Support-dependent envelope and influence

Fix integers $K \geq 1$ and $1 \leq m \leq K$. For $S \in \{0, \ldots, K\}^B$, let $C(S) = \sum_i \mathbf{1}\{S_i \geq m\}$, $E(S) = \{i : S_i > 0\}$, and $A_m(S)$ be the allocation in Eq. equation 3, with its stated boundary conventions.

Proof of Theorem 1: envelope. For $0 < C < n$, evaluate the mass function at $1/m$:

$$f_S(1/m) = \sum_i \min(1, S_i/m) \geq \sum_i \mathbf{1}\{S_i \geq m\} = C.$$

By monotonicity, the root of f_S is at most $1/m$, which gives $q_i \leq \min(1, S_i/m)$. If $C = 0$, all weights are zero. If $C = n > 0$, every positive score is at least m and every eligible weight is one, so the bound still holds. In this case, $\lambda_0 = 1/\min_{i \in E} S_i \leq 1/m$ is a valid choice of root. The bound concerns the allocation itself; an implementation may return a different multiplier when all eligible weights are saturated. □

Proof of Theorem 1: a single judgment. Let $S' = S + \mathbf{e}_j$ and write $q = A_m(S)$, $q' = A_m(S')$. Exactly one count increases, so C' equals C or $C + 1$.

Unchanged total. If $C = 0$, both allocations are zero. Otherwise, choose a root for the original allocation satisfying $\lambda_0 \leq 1/m$, as above. Increasing one score increases the mass function pointwise. We can therefore choose a root for the new allocation with $\lambda' \leq \lambda_0$. At every coordinate whose score is unchanged, $q'_i \leq q_i$. Since the totals are equal, $q'_j \geq q_j$ and

$$\|q' - q\|_1 = 2(q'_j - q_j).$$

The function $z \mapsto \min(1, z)$ is nondecreasing and one-Lipschitz, giving

$$q'_j - q_j \leq \min(1, \lambda_0(S_j + 1)) - \min(1, \lambda_0 S_j) \leq \lambda_0 \leq 1/m.$$

The argument also covers a newly eligible coordinate and an original solution in which all eligible weights are saturated. If all eligible weights in the new solution are saturated at the same total, both the eligible set and the allocation are unchanged.

Threshold crossing. Here $S_j = m - 1$ and $C' = C + 1$. If $C = 0$, the original allocation is zero and the claim follows immediately. Otherwise, at a root for the original allocation,

$$f_{S'}(\lambda_0) - f_S(\lambda_0) = \min(1, \lambda_0(S_j + 1)) - \min(1, \lambda_0 S_j) \leq 1.$$

Thus $f_{S'}(\lambda_0) \leq C + 1$. If the new solution has an unsaturated coordinate, its unique root is at least λ_0, so no weight decreases. If all eligible coordinates are saturated, the same conclusion follows from $E \subseteq E'$. Therefore

$$\|q' - q\|_1 = \sum_i (q'_i - q_i) = C' - C = 1.$$

Changing a favorable judgment to an unfavorable one reverses one of these comparisons. The argument uses no distributional assumptions on the judgments. □

Interpretation and a threshold comparison. The support envelope also holds for any prescribed total no larger than $\sum_i \min(1, S_i/m)$. SPG's choice of C satisfies this condition while allowing contexts below the threshold to receive weight up to S_i/m.

For $m \geq 2$, consider exchanging two scores $(m, m-1)$. The score multiset, total, and root remain unchanged, so the distance between the allocations is at most $2/m$. The binary rule $\mathbf{1}\{S_i \geq m\}$ instead switches between $(1, 0)$ and $(0, 1)$, a distance of two. Comparing these rules at the same total isolates the smoother response of proportional allocation near the threshold.

Propagation to the actor gradient. Let $\widehat{\mathcal{L}}_{\mathrm{RL},i}$ and $\widehat{\mathcal{L}}_{\mathrm{P},i}$ be the per-context sampled losses defined in Section 4. When the references, actor noise, and these loss gradients are fixed, stopped-gradient allocation gives

$$g(q) = g_{\mathrm{RL}} + \frac{1}{B}\sum_i q_i \delta_i, \qquad g_{\mathrm{RL}} = \frac{1}{B}\sum_i \nabla \widehat{\mathcal{L}}_{\mathrm{RL},i}, \quad \delta_i = \nabla \widehat{\mathcal{L}}_{\mathrm{P},i} - \nabla \widehat{\mathcal{L}}_{\mathrm{RL},i}. \tag{A.3}$$

Consequently, for $D = \max_i \|\delta_i\|_2$,

$$\|g(q') - g(q)\|_2 \leq \frac{1}{B}\sum_i |q'_i - q_i| \|\delta_i\|_2 \leq \frac{D}{B}\|q' - q\|_1. \tag{A.4}$$

Equation equation A.4 isolates the effect of changing the weights, whether through an evidence judgment or numerical rounding. Resampling subgoals or actor actions also changes the target or per-context loss gradients and is therefore a different comparison.

A.3 Relative evidence and an actual optimizer step

Fix the training state immediately before the actor update and the random samples used to compute the per-context losses. For $n > 0$, let $q = A_m(S)$, $b = (C/n)\mathbf{1}_E$, $d = q - b$, and $\widehat{p}_i = S_i/K$ on E. Define $H = d^\top \widehat{p}$ and $R = \|d\|_2$. All vectors are zero outside E. Since q_i is nondecreasing in $\widehat{p}_i$,

$$H = \frac{1}{2n} \sum_{i,j \in E} (q_i - q_j)(\widehat{p}_i - \widehat{p}_j) \geq 0.$$

The identity follows by expanding the double sum and using $\sum_i q_i = C$.

Let $g(q)$ and δ_i be defined by Eq. equation A.3. For a differentiable evaluation objective J, let $\mathsf{T}_\nu(g)$ be one step of the actual optimizer from fixed state ν, and write $F_\nu = J \circ \mathsf{T}_\nu$. Define $u_i = \langle \nabla F_\nu(g(b)), \delta_i \rangle$. Then u_i/B is the first-order evaluation effect per unit of context i's transfer coefficient. The following comparison permits a positive association with support counts and a bounded residual.

Proposition A.2 (Conditional comparison with uniform allocation)**.** *Suppose $n > 0$, F_ν has an L_ν-Lipschitz gradient on a neighborhood of the segment from $g(b)$ to $g(q)$, and on E, $u_i = a + \beta \widehat{p}_i + e_i$ with $\beta > 0$ and $\|e\|_2 \leq \epsilon_u$. Then*

$$J(\mathsf{T}_\nu(g(q))) - J(\mathsf{T}_\nu(g(b))) \geq \frac{\beta H - \epsilon_u R}{B} - \frac{L_\nu}{2} \|g(q) - g(b)\|_2^2. \tag{A.5}$$

Proof of Proposition A.2. Write $g_b = g(b)$, $g_q = g(q)$, and $z = g_q - g_b = B^{-1} \sum_i d_i \delta_i$. By the assumed Lipschitz continuity of ∇F_ν,

$$F_\nu(g_q) - F_\nu(g_b) \geq \langle \nabla F_\nu(g_b), z \rangle - \frac{L_\nu}{2} \|z\|_2^2.$$

By the definition of u_i, the first term equals $B^{-1} \sum_i d_i u_i$. Substituting $u_i = a + \beta \widehat{p}_i + e_i$ gives

$$\sum_i d_i u_i = a \sum_i d_i + \beta d^\top \widehat{p} + d^\top e = \beta H + d^\top e \geq \beta H - \epsilon_u R,$$

where the intercept cancels because the allocations have the same total, and the last step follows from Cauchy–Schwarz. Since $F_\nu(g) = J(\mathsf{T}_\nu(g))$, combining these inequalities proves Eq. equation A.5. If $q = b$, the gradients and parameter updates coincide, and every term is zero. □

The linear mismatch penalty is tight: for $R > 0$, choosing $e = -\epsilon_u d/R$ gives $d^\top e = -\epsilon_u R$. Support therefore need not rank every sample correctly, provided its aggregate signal exceeds the mismatch. If no coordinate is saturated and $C > 0$, setting $\bar{p} = n^{-1} \sum_i \widehat{p}_i$ gives

$$H = \frac{C}{\bar{p}} \operatorname{Var}_E(\widehat{p}), \qquad R = \frac{C}{\bar{p}} \sqrt{\frac{\operatorname{Var}_E(\widehat{p})}{n}}.$$

The linearized gain is therefore strictly positive whenever $\beta \sqrt{n \operatorname{Var}_E(\widehat{p})} > \epsilon_u$ and $\operatorname{Var}_E(\widehat{p}) > 0$. The curvature term in Proposition A.2 accounts for the finite step.

The fixed-state map includes Adam. For a standard Adam step at counter t with previous moments M, V, learning rate h, decay factors β_1, β_2, and denominator constant $\varepsilon_A > 0$, write

$$A_j = \frac{\beta_1 M_j}{1 - \beta_1^t}, \quad B_1 = \frac{1 - \beta_1}{1 - \beta_1^t}, \quad D_j = \frac{\beta_2 V_j}{1 - \beta_2^t}, \quad D_2 = \frac{1 - \beta_2}{1 - \beta_2^t}.$$

Its coordinate map is

$$[\mathsf{T}_\nu(g)]_j = \theta_j - h \frac{A_j + B_1 g_j}{\sqrt{D_j + D_2 g_j^2} + \varepsilon_A}.$$

The map includes the current gradient's contribution to Adam's second moment (Kingma & Ba, 2015). Positive D_j ensures local smoothness of the optimizer map. Proposition A.2 additionally requires the composite F_ν to have a locally Lipschitz gradient along the comparison segment. If some D_j vanish, the same local regularity condition must still be checked.

The comparison concerns the chosen objective J: expected environment return when J is that return, or the specified surrogate otherwise. An empirical test of the support–utility relation should assess utility independently of the critic used to form S_i, avoiding circular validation.

A.4 Density coverage of the complete planning mixture

Fix a context and write π_k for its kth subgoal-conditioned component. Then $\pi_{\text{plan}} = K^{-1} \sum_k \pi_k$. Retaining all components gives the following relation to any reduced mixture.

Lemma A.3 (Full-mixture density coverage)**.** *Let $I \subseteq \{1, \dots, K\}$ be nonempty, let $M = |I|$, and define $\pi^I_{\text{plan}} = M^{-1} \sum_{k \in I} \pi_k$. Suppose all distributions have densities relative to a common dominating measure. If $D_{\text{KL}}(\pi \| \pi^I_{\text{plan}}) < \infty$, then*

$$D_{\text{KL}}(\pi \| \pi_{\text{plan}}) \le D_{\text{KL}}(\pi \| \pi^I_{\text{plan}}) + \log(K/M). \tag{A.6}$$

Proof. Pointwise,

$$\pi_{\text{plan}} = \frac{1}{K} \sum_{k=1}^{K} \pi_k \ge \frac{M}{K} \pi^I_{\text{plan}}.$$

Finite $D_{\text{KL}}(\pi \| \pi^I_{\text{plan}})$ implies $\pi^I_{\text{plan}} > 0$ almost surely under π. Taking logarithms of the pointwise inequality and integrating proves Eq. equation A.6. □

The full reference retains each subset's density with at least that subset's mixture weight.

B Implementation and Reproducibility Details

B.1 Reference construction and differentiation

For learned-subgoal guidance, a Laplace planner provides one pool of $K = 16$ subgoals per replay context. The target actor is a diagonal Gaussian; its subgoal-conditioned densities form the reference mixture, and samples from those densities supply the evidence actions. The online actor draws a pre-tanh action v_i and executes $a_i = \tanh(v_i)$. Thus critic comparisons use sampled actions, whereas the KL term evaluates the component densities.

In pre-tanh space, use the superscript v for densities and abbreviate the context-conditioned reference as $\pi^v_{\text{plan},i}$. The reference and Monte Carlo KL contribution are

$$\pi^v_{\text{plan},i}(v) = \frac{1}{K} \sum_k \mathcal{N}(v; \text{sg}(\mu_{ik}), \text{sg}(\sigma_{ik})^2),$$
$$\widehat{D}_i = \log \pi^v_\theta(v_i \mid x_i) - \log \pi^v_{\text{plan},i}(v_i), \qquad v_i \sim \pi^v_\theta(\cdot \mid x_i).$$

Here $\text{sg}(\sigma_{ik})^2$ specifies the diagonal variances. Because actor and reference share the invertible tanh transform, their Jacobian terms cancel in the log-density ratio. We sum component log densities over action dimensions, apply log-sum-exp across components, and subtract $\log K$. The KL divergence is nonnegative, but an individual Monte Carlo contribution need not be.

We detach the reference means and scales while retaining gradients through v_i in both log densities. With the computed allocation held fixed, the sampled actor loss is

$$\frac{1}{B} \sum_i \left[-(1 - q_i) \min_h Q_{\phi,h}(s_i, a_i, g_i) + q_i \alpha \widehat{D}_i \right].$$

The evidence and both losses reuse the same actor draw. Support scores, threshold decisions, the normalizer, and allocation weights are detached. The resulting gradient is conditional on this draw and allocation; it does not differentiate through the allocation's dependence on the actor. It is the gradient $g(q)$ used in the fixed-state analysis, rather than the full derivative of a population objective that includes that dependence.

B.2 Network roles and update order

Each learned-subgoal learner update trains the online twin critics and subgoal model before computing support and updating the actor. Target networks are updated last. Critic bootstrapping evaluates an

online-actor sample with the target twin-critic minimum. The fixed-state comparisons in Section 4 begin immediately before the actor step, after the critic and planner have been updated.

For evidence, the implementation computes the mean of the target heads minus their population standard deviation, with coefficient one. With two heads this equals their minimum in exact arithmetic, although the floating-point code retains the mean-minus-standard-deviation form. The return loss uses the online critic.

The planner keeps its value-weighted log-likelihood objective and softmax temperature. The multiplier λ is recomputed for each batch and belongs only to the allocation solver. During actor differentiation, target parameters remain fixed. No weighting network or separate optimizer is introduced.

B.3 Configuration and initialization

Table 2 gives the learned-subgoal implementation, and Table 3 specifies the control-study evaluation protocol. The six-task evaluation is described in Appendix C.1.

Table 2: Learned-subgoal implementation settings.

Quantity	Value or behavior
Support count / support fraction	$K = 16$, $\eta = 0.25$, $m = 4$
Transfer scale / batch size	$\alpha = 0.1$, $B = 2048$
Actor / critic / planner hidden sizes	two layers of 256 ReLU units
Actor / critic / planner learning rates	10^{-3} / 10^{-3} / 10^{-4}
Discount / target Polyak coefficient	0.99 / 0.005
Evidence comparison	strict positive target-critic difference
Initial random interactions	10,000 environment steps
Common transfer-only warmup	5,000 learner updates
Active-phase allocation	capped proportional, no sampling
Optimizer	inherited Adam moments and counters
Gradient clipping / extra actor entropy	absent / absent
Solver / training weight dtype	float64 / float32

After 10,000 random interactions, each environment step is followed by one learner update. Updates 0–4999 use the common transfer-only warmup ($q_i = 1$). Active allocation begins at learner update 5000—environment step 15001 in the implementation's zero-based loop—and inherits the Adam moments from warmup. The allocation theorems apply from this point onward. Deployment follows each planning source's original protocol, with no inference-time module added by SPG.

B.4 Numerics and solver checks

The allocator sorts positive counts, computes reverse cumulative sums in float64, and applies Eq. equation A.2. It returns zero for $C = 0$ and one on E for $C = n$; the returned multiplier need not be the smallest valid root in these boundary cases. The float64 total must be within 10^{-10} of C. After casting to float32, the allowed absolute error is 2×10^{-4} at the stated batch size. Allocation is deterministic. Equation equation A.4 bounds the gradient error caused by rounding the weights.

Implementation checks cover boundary cases, total-weight conservation, and the actor-gradient interface. The fixed-state analysis in Appendix C.5 evaluates the support envelope and single-judgment influence at stored training states. Supplementary scripts reproduce the numerical summaries from the per-state measurements; the available tensor-level artifacts are specified in that appendix.

C Experimental Protocols and Additional Results

The evaluation combines six-task learning curves, paired allocation comparisons on two navigation layouts, and fixed-state and trajectory analyses. The following sections define their respective protocols, statistical units, and scope.

C.1 SIX-TASK EVALUATION

Figure 3 reports the RIS-planner comparisons on six tasks, with budgets of 300K interactions on AntMaze-U and S and 400K on the remaining tasks. The main evaluation reports four seeds and standard-deviation bands, as specified in Section 5.1. Oracle replaces only the support-evaluation critic within SPG. Figure 4(a)–(b) separately evaluates graph-derived planning on AntMaze-U and W at their respective budgets.

The per-seed tables below belong to a separate five-seed allocation study. Its random/farthest joint-success metric and integration windows are defined in Section C.2; they are not the six-task random-goal curves or the fixed-goal allocation comparison in Figure 5. In particular, its AntMaze-U budget is 150K rather than 300K. These results retain their original statistical units and are not pooled with the main evaluation.

C.2 CONTROLLED-STUDY PROTOCOL

The control study comprises 40 completed configuration–seed runs: two layouts, four allocation rules, and five training seeds. Both layouts use the online learned Laplace subgoal model and the actor–critic specified in Appendix B. Training starts with 10,000 random interactions, followed by 5,000 common transfer-only learner updates. The active allocation rules then operate with the inherited networks and optimizer states. The support pool is sampled once and reused for the evidence calculation and the full-mixture transfer target.

Table 3: **Protocol of the paired allocation study.** In the interaction-budget rows, K denotes thousands; the candidate support size is stated separately.

Setting	AntMaze-U	AntMaze-W
Environment registration	`AntMaze-U` family	`AntMaze-WMed` family
Training budget	150K interactions	400K interactions
Primary AUC window	50K–150K	50K–400K
Exported evaluation grid	15K:5K:150K	15K:5K:400K
Random / farthest evaluation episodes	100 / 20	100 / 20
Evaluation horizon / success radius	600 / 0.5	600 / 0.5
Training seeds per allocation rule	5	5
Support size / support fraction	16 / 0.25	16 / 0.25
Transfer scale / replay batch size	0.1 / 2048	0.1 / 2048
Fixed eligible coefficient q_0	0.768071461651	0.771510797297

The environment registrations are `AntMaze-UTrainEnv-v0` and the `AntMaze-WMedTrain Env-v1` family with their corresponding test environments. Five seeds are shared across methods and layouts: 409440170, 104058272, 1714577578, 1060967083, and 1930300609. Shared seeds define the pairing, but do not imply that separately learned policies encounter identical subsequent trajectories or support budgets.

For each checkpoint, we first compute

$$y_{\text{joint}}(t) = \min\{y_{\text{random}}(t), y_{\text{farthest}}(t)\}.$$

Each primary AUC then uses the trapezoidal rule on the complete common grid:

$$\text{AUC}(y; t_0, T) = \frac{1}{T - t_0} \sum_{j=0}^{M-1} (t_{j+1} - t_j) \frac{y(t_j) + y(t_{j+1})}{2}, \qquad t_0 = 50{,}000. \tag{C.1}$$

The minimum is taken before integration. Terminal success is the last checkpoint value, and trailing success averages the last five observed checkpoints. The 40 per-seed AUC values are computed from 2,120 checkpoint observations on the stated grids. Curves are unsmoothed and end at their respective training budgets; missing outcomes are not imputed.

C.3 WHAT THE CONTROLS CHANGE

All four numerical-study rules use the same full-K reference and the complementary actor loss in Eq. equation 4. Their coefficient vectors differ as follows:

$$q_i^{\mathrm{SPG}} = \min\{1, \lambda S_i\}\mathbf{1}\{i \in E\}, \qquad \sum_i q_i^{\mathrm{SPG}} = C,$$

$$q_i^{\mathrm{UE}} = (C/n)\mathbf{1}\{i \in E\}, \qquad \sum_i q_i^{\mathrm{UE}} = C,$$

$$q_i^{\mathrm{UB}} = C/B, \qquad \sum_i q_i^{\mathrm{UB}} = C,$$

$$q_i^{\mathrm{FE}} = q_0\mathbf{1}\{i \in E\}, \qquad \sum_i q_i^{\mathrm{FE}} = q_0 n.$$

For $n = 0$, the eligible-set allocations are zero. UE denotes uniform on the eligible set, UB uniform over the batch, and FE the fixed eligible coefficient.

UE differs from SPG only in relative shares when compared at the same prestate, support, and sampled losses. UB additionally gives transfer weight to contexts with no favorable evidence. FE changes both the amount and the shares relative to SPG; it does not retain the proportional solver. Its two layout-specific constants were fixed using a separate development cohort before these control results. FE is consequently a development-calibrated constant-coefficient comparison, not an isolated test of instantaneous budget adaptation.

In the six-task comparison, RIS denotes conventional planning regularization, SPReD the evaluated uncertainty-weighted adaptation, and PCGrad gradient-level handling of the objectives. These are different comparisons from UE, UB, and FE. In particular, conventional additive regularization averages $\mathcal{L}_{\mathrm{RL},i} + \mathcal{L}_{\mathrm{P},i}$, whereas the controlled allocations use complementary coefficients. Their updates are not interchangeable; Appendix C.1 states the available implementation detail for the main baselines.

C.4 ALLOCATION PROFILES AND TRAINING DYNAMICS

The fixed-state analysis applies SPG to twenty actor-update states from the UE control: one 2,048-context batch at 50K and 100K on AntMaze-U, and at 50K and 200K on AntMaze-W, for each seed. These states support the coefficient and fixed-gradient comparisons. Training dynamics are measured separately on ten SPG trajectories, with 2,600 batch-level observations at 1K-interaction intervals. Row-level SPG trajectories are unavailable, so the support-conditioned profile uses the UE states rather than SPG's state distribution.

Table 4 exhibits graded shares below saturation rather than a binary threshold. Its context counts and saturation frequencies describe the fixed UE states. They are not estimates of SPG's visitation distribution, and they do not increase the number of independent training runs.

The largest observed total-weight residual is 1.59145×10^{-5} after casting to the training dtype, below the 2×10^{-4} tolerance. The measurements show broad but partial transfer at a relatively stable mean guidance level. They establish total-weight behavior, but do not identify temporal adaptation as the principal source of the learning gain.

Table 4: **SPG weights evaluated on stored control-policy states.** Descriptive summaries across 40,960 contexts, after applying the SPG operator. Quantiles are not confidence intervals. All envelope residuals are zero at tolerance 10^{-12}.

S_i	Contexts	Mean q_i	5th pct.	95th pct.	Max. q_i	Upper bound
0	2970	0.0000	0.0000	0.0000	0.0000	0.00
1	2886	0.1421	0.1384	0.1461	0.1465	0.25
2	2810	0.2843	0.2768	0.2922	0.2930	0.50
3	2646	0.4262	0.4152	0.4383	0.4395	0.75
4	2728	0.5689	0.5607	0.5860	0.5860	1.00
5	2596	0.7113	0.6986	0.7325	0.7325	1.00
6	2646	0.8535	0.8410	0.8766	0.8790	1.00
7	2467	0.9905	0.9688	1.0000	1.0000	1.00
8	2501	1.0000	1.0000	1.0000	1.0000	1.00
9	2389	1.0000	1.0000	1.0000	1.0000	1.00
10	2306	1.0000	1.0000	1.0000	1.0000	1.00
11	2197	1.0000	1.0000	1.0000	1.0000	1.00
12	2172	1.0000	1.0000	1.0000	1.0000	1.00
13	2043	1.0000	1.0000	1.0000	1.0000	1.00
14	1975	1.0000	1.0000	1.0000	1.0000	1.00
15	1875	1.0000	1.0000	1.0000	1.0000	1.00
16	1753	1.0000	1.0000	1.0000	1.0000	1.00

Table 5: **Periodic telemetry on actual SPG trajectories.** Each metric is averaged within a seed, then across five seeds (mean $\pm$ sample SD). $N_{\text{eff}} = (\sum_i q_i)^2 / \sum_i q_i^2$ measures concentration of coefficient mass, not independence of replay samples.

Layout	Records	C/B	n/B	N_{eff}/B
AntMaze-U	675	0.7050 ± 0.0037	0.9201 ± 0.0020	0.7947 ± 0.0030
AntMaze-W	1925	0.7117 ± 0.0016	0.9228 ± 0.0007	0.7998 ± 0.0012

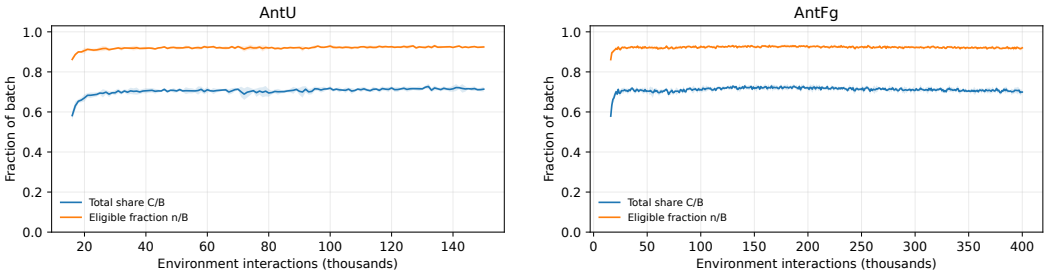


Figure 7: **Actual guidance dynamics.** Periodic C/B and n/B measurements from SPG runs, with mean and sample-SD bands over training seeds. Each point summarizes the recorded 1K-interaction sampling cadence.

C.5 Single-judgment perturbations and optimizer response

At each of the 20 stored states, we evaluate 128 unfavorable-to-favorable and 128 favorable-to-unfavorable single-candidate changes. Each change triggers recomputation of both the strict count C and the proportional allocation, with the planning reference and actor sample held fixed. These 5,120 fixed-state counterfactuals isolate allocation response without additional environment interactions.

All allocation residuals fall within the 10^{-12} tolerance. Threshold crossings have maximum excess over one of 9.24×10^{-14}, and the direction-specific monotonicity check is satisfied. The upper bound is not expected to be attained in each state: many changed votes leave a saturated sample's allocation unchanged or induce only a small reallocation among the unsaturated samples.

Table 6: **Single-judgment effects.** Distances are $\|q' - q\|_1$. The maximum in the crossing group is one up to floating-point roundoff.

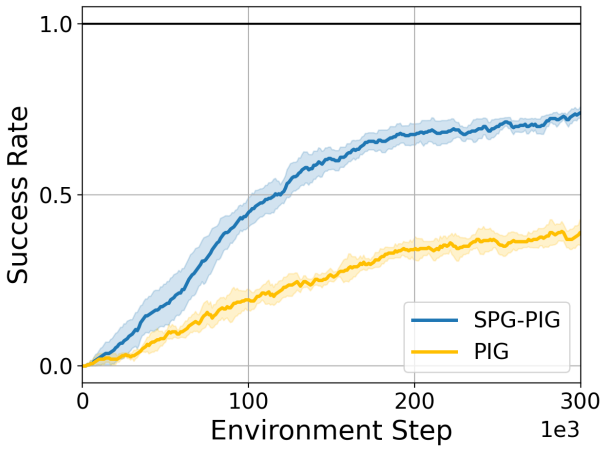
Regime	Changes	Median	95th pct.	Maximum	Bound
Total unchanged	4,771	0.01247	0.29099	0.29301	0.5
Threshold crossed	349	1.00000	1.00000	1.00000	1.0

The 80 gradient/optimizer comparisons use the same parameters, online critic, full planning reference, actor noise, and inherited Adam moments for the original and perturbed allocations. The stored per-row gradient differences provide $D = \max_i \|\delta_i\|_2$, so the comparison checks the bound in Eq. equation A.4 directly. No violation exceeds the 10^{-9} bound tolerance. The median Adam-parameter displacement difference is 4.73228×10^{-5}. Adam is replayed from a common state for each alternative, rather than applied sequentially or with reset moments.

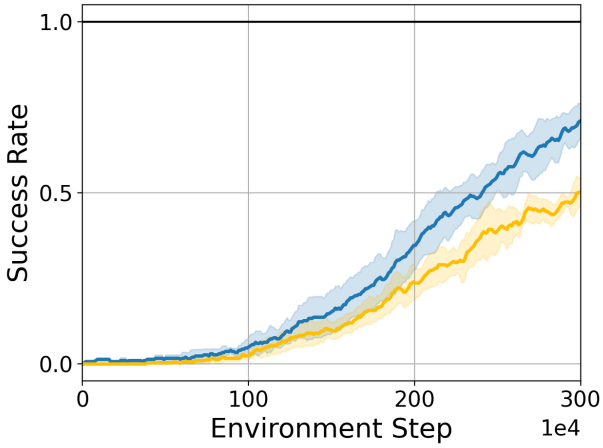
The supplementary data contain the per-state numerical diagnostics and their factorization, replay, and optimizer-parity tolerances. The underlying prestate tensors are not included. Consequently, the analysis scripts reproduce the aggregate statistics, while independent repetition of the neural-gradient calculations requires those states.

C.6 SPG-PIG RESULTS

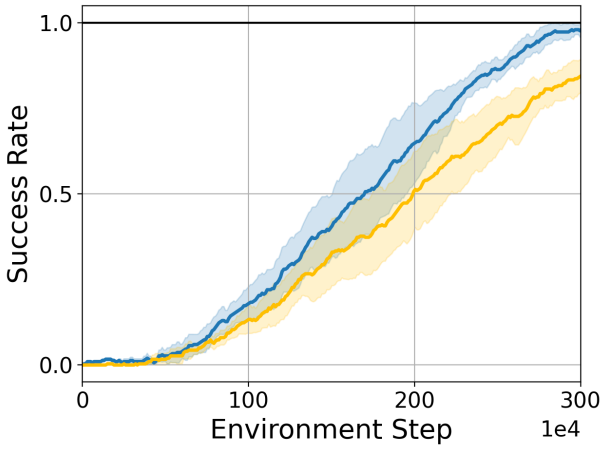
Figure 8 compares SPG-PIG and PIG on two navigation layouts and one manipulation task.



(a) AntMaze-U



(b) AntMaze-W



(c) FetchPush

Figure 8: **SPG-PIG experimental results.** Success rate versus environment interactions for SPG-PIG and PIG on AntMaze-U, AntMaze-W, and FetchPush. Curves and shaded regions show the reported aggregate trajectories and their variability.