Budgeted Human Steering for Long-Horizon Agents via Adaptive Intervention Allocation

Nathon Lee
leejianwoo@gmail.com

Atlas Morgan
hitomli@163.com

June 2026

Abstract

Long-horizon agents frequently fail due to sparse critical decisions where small mistakes compound. Human guidance can correct such decisions, but human attention is limited and expensive. We study the problem of budgeted human steering, where a teacher can intervene at most B times per trajectory. The central question is whether *when* to spend the intervention budget matters more than the total budget magnitude.

We implement a prototype system with an adaptive budget controller, local teacher interventions, and offline/online policy distillation. Evaluation on a calibrated long-horizon toy environment (v3 medium) reveals: (1) one adaptive intervention improves success from 3% to 62%, matching the performance of eight forced interventions; (2) timing-aware allocation substantially outperforms random and forced spending strategies; (3) behavior cloning on corrected actions outperforms KL distillation to teacher logits; (4) online policy rounds provide no advantage over offline replay in this stationary setting. These results demonstrate that intervention timing is critical, yet converting additional interventions into robust policy improvement remains an open challenge. We present this work as a reproducible prototype and diagnostic benchmark for budgeted steering research.

1 Introduction

Deep reinforcement learning agents have achieved impressive performance on many control tasks, yet long-horizon environments remain challenging. A key difficulty is that task success often depends on a small number of critical decisions: a mistake at any critical moment may doom the entire trajectory, even if the agent recovers partially. This motivates studying how to guide agents through rare but consequential decisions.

Human intervention provides a natural solution. A teacher observing the agent's behavior can identify when the agent is about to make a critical mistake and provide a correction. However, human attention is a scarce resource. In practice, a teacher might have time to intervene only a handful of times per episode. This motivates the budgeted steering problem: given a per-episode budget of B interventions, how should the teacher allocate that budget to maximize the agent's success rate?

The natural assumption might be that more budget is always better. However, our experiments reveal a more nuanced picture. The key insight is that *when* the intervention occurs matters as much as the total number of interventions. A single well-timed intervention at a critical moment can teach the agent a reusable local correction, whereas eight interventions spent uniformly across the episode are nearly useless. This is fundamentally a problem of intervention *targeting*, not merely sample efficiency.

The effectiveness of a single intervention stems from the structure of long-horizon tasks: success often depends on a small number of critical decision nodes. In our environment, only 6 critical nodes exist across a 32-step trajectory, and the agent must pass 4 of them. An untrained agent succeeds with probability roughly $\binom{6}{4}p^4(1-p)^2$ where $p \approx 0.25$ is the baseline per-node success rate, yielding $\approx$3% overall success. A single well-placed intervention that guarantees one node's success changes the requirement to passing 3 out of 5 remaining nodes, dramatically increasing the probability to $\approx$62%. Moreover, behavior cloning on corrected actions allows the student to extract generalizable decision rules—not just memorize a single state-action pair, but learn patterns such as "when the critical flag is set and risk is high, take the teacher's action." This learned rule transfers to other critical moments the teacher never directly corrected.

We formalize budgeted steering with an adaptive budget controller that prioritizes critical states, implements local teacher interventions, and uses behavior cloning to distill corrected actions into a student policy. We also explore optional KL distillation to teacher logits and online policy aggregation. Our experiments on a calibrated toy environment reveal strong performance of adaptive timing but also honest limitations: additional budget yields diminishing returns, KL distillation does not help, and online methods do not outperform offline training in stationary settings.

Our contributions are: (1) a formal problem formulation for budgeted human steering; (2) a simple prototype combining adaptive allocation, local interventions, and policy distillation; (3) a calibrated toy environment that avoids ceiling and floor effects; (4) systematic ablations showing that timing dominates budget magnitude; (5) transparent reporting of negative findings and mechanistic analysis of why KL distillation fails in this setting.

This work is positioned as a prototype study and diagnostic benchmark rather than a complete solution. We hope the reproducible setup and honest empirical results will guide future research on sample-efficient human-steered learning.

2 Related Work

Our work intersects several research areas: human feedback in learning, imitation learning, policy distillation, and sample-efficient reinforcement learning.

2.1 Human Feedback and Interactive Learning

Early work by Knox and Stone [2008] introduced TAMER, where agents learn from human evaluative feedback by associating observed trajectories with human reward signals. More recently, Christiano et al. [2017] proposed learning reward functions from human preferences collected via pairwise comparisons, enabling deep RL from human feedback (RLHF) at scale. Warnell et al. [2018] extended TAMER to high-dimensional state spaces using deep neural networks. Unlike these preference-based approaches, our work focuses on sparse action-level interventions where a teacher directly corrects an agent's proposed action rather than providing scalar reward or ranking trajectories.

2.2 Imitation Learning and Dataset Aggregation

Ross et al. [2011] introduced DAgger (Dataset Aggregation), a foundational algorithm for online imitation learning that alternates between collecting trajectories with a learned policy and training on the oracle's corrections. DAgger solves the covariate shift problem by ensuring the training distribution aligns with the policy distribution. Our work also uses oracle corrections, but under a budget constraint—the teacher cannot answer every query, so the problem becomes *when to query*.

More broadly, Abbeel and Ng [2004] formalized apprenticeship learning as recovering a reward function from demonstrations, while Ziebart et al. [2008] proposed maximum entropy inverse RL, which avoids reward ambiguity through entropy regularization.

2.3 Learning from Demonstrations and Sparse Interventions

Hester et al. [2018] proposed Deep Q-Learning from Demonstrations (DQfD), which initializes a policy from a small set of demonstrations and uses them to accelerate learning. Brown et al. [2019] studied learning from demonstrations in complex robotic navigation tasks where human demonstrations provide sparse supervision. More recently, Mandlekar et al. [2021] examined representation learning for robotic tasks using self-supervised and supervised signals. These works treat demonstrations as a form of scarce, expensive supervision. Our approach is conceptually similar: interventions are sparse corrections that must be used wisely.

2.4 Policy and Knowledge Distillation

Hinton et al. [2015] introduced knowledge distillation for neural networks, where a small "student" network is trained to mimic a large "teacher" network by matching logit distributions via KL divergence. Rusu et al. [2015] adapted this idea to reinforcement learning, proposing policy distillation to transfer knowledge between policies. In our work, we explore both behavior cloning (directly matching teacher actions) and KL distillation (matching teacher logits). We find that BC outperforms KL in this setting, suggesting that teacher logit alignment may not always be beneficial. We investigate the mechanistic reasons for this failure in Section 7.2.

2.5 Deep Reinforcement Learning Fundamentals

The foundations of deep RL come from Mnih et al. [2015], who introduced Deep Q-Networks (DQN) and demonstrated superhuman performance on Atari games. Schulman et al. [2017] later proposed Proximal Policy Optimization (PPO), a more sample-efficient and stable policy gradient algorithm. These methods form the conceptual basis for the student policy learning in our work.

2.6 Sample Efficiency and Rapid Adaptation

Sample efficiency is critical when human intervention is costly. Finn et al. [2017] proposed Model-Agnostic Meta-Learning (MAML), enabling rapid adaptation to new tasks with few gradient steps. Chua et al. [2018] achieved sample-efficient model-based RL using learned dynamics models and probabilistic planning. Kumar et al. [2020] introduced Conservative Q-Learning (CQL) for offline RL, which prevents extrapolation errors by constraining learned Q-values to regions supported by the dataset. Nagabandi et al. [2019] extended meta-learning to adapt to non-stationary environments. These works emphasize that learning efficiency improves when methods avoid distributional mismatch and adapt quickly to available data.

2.7 Offline and Online Learning Trade-offs

Levine et al. [2020] provided a comprehensive review of offline RL methods, which learn from fixed datasets without online environment interaction. The trade-off between offline stability and online adaptivity has motivated hybrid approaches. Ho and Ermon [2016] introduced Generative Adversarial Imitation Learning (GAIL), which learns reward functions and policies from demonstrations

using adversarial training. We explore both offline training from collected interventions and online rounds that alternate collection and training; neither shows a clear advantage in stationary settings.

2.8 Uncertainty and Risk-Driven Interventions

Recent work has emphasized the role of uncertainty in guiding exploration and intervention. Kendall and Gal [2017] analyzed different types of uncertainty (aleatoric and epistemic) in deep learning for vision tasks, providing tools to estimate when a model is uncertain. Uncertainty estimates could in principle drive intervention decisions, but our current prototype uses environment probe information as a proxy for risk.

2.9 Representation Learning and Transfer

Pathak et al. [2016] proposed context encoders for unsupervised representation learning, showing that predicting missing patches in images learns useful visual features. Laskey et al. [2017] studied domain randomization and representation learning for robotic control. These works suggest that good state representations improve sample efficiency, relevant to long-horizon learning where the state space is large.

3 Problem Formulation

We consider an episodic reinforcement learning setting with horizon H. At each step $t \in \{0, 1, \ldots, H-1\}$, the agent observes state s_t, proposes action $a_t \sim \pi_\theta(s_t)$, and receives the environment's response (next state s_{t+1} and reward r_t).

A teacher or human oracle has access to the true optimal action a_t^* at each step (or at least can recognize when a_t is suboptimal). The teacher can provide an intervention e_t that corrects the agent's action. However, the teacher's attention is limited: the teacher can provide at most B interventions per episode, where B is a budget parameter.

The problem is to design:

1. An adaptive **budget controller** that decides *when* to spend an intervention (given remaining budget and trajectory state),
2. A **teacher intervention** mechanism that corrects the agent's action,
3. A **policy distillation** objective that learns from the collected interventions.

The goal is to maximize success rate on autonomous evaluation (after training, with no interventions).

3.1 Budget Constraint

Per episode, the cumulative cost of interventions must satisfy:

$$\sum_{t=0}^{T-1} \mathbb{1}[\text{intervened at step } t] \leq B$$

where T is the actual episode length (possibly shorter than H if the episode terminates early).

3.2 Intervention Event

An intervention event e_t at step t contains:

- **Corrected action**: a_t^* (the oracle's action, or a correction signal),
- **Teacher logits** (optional): $p_{\text{teacher}}(\cdot|s_t)$, the teacher's action distribution,
- **Local effect**: the intervention corrects the current step and may provide a short follow-up advantage (e.g., suppressed transition noise for span steps).

Unlike full demonstrations, interventions are local: they fix the current decision but not the entire trajectory.

4 Method

4.1 Adaptive Budget Controller

The budget controller decides whether to intervene at step t based on a scoring function. We use:

$$\text{score}_t = w_u \cdot U_t + w_h \cdot \frac{t}{H} + w_c \cdot \mathbb{1}[\text{critical}_t] + w_r \cdot \text{risk}_t \tag{1}$$

where:

- U_t: uncertainty or confidence estimate of $\pi_\theta(s_t)$,
- t/H: horizon pressure (encourage spending budget before late stages),
- $\mathbb{1}[\text{critical}_t]$: indicator of critical state (task-specific),
- risk_t: estimated risk or importance (how much the agent's decision matters),
- w_u, w_h, w_c, w_r: learned or hand-tuned weights.

The controller intervenes if $\text{score}_t \geq \tau$ (threshold) and budget remains.

In the current prototype, we use environment probe information to compute criticality and risk in a controlled setting. This allows us to isolate the value of timing-aware allocation without conflating it with the difficulty of learning to detect criticality. Real deployments would require learning to estimate these quantities from observations and uncertainty estimates.

4.2 Teacher Intervention

At each critical state, the teacher provides the oracle action or signals an action veto if the agent's proposed action is a trap. The intervention suppresses transition noise for one step, ensuring the agent's action leads to the expected outcome. The intervention also seeds a local follow-up advantage (via intervention_effect_span), allowing the agent to exploit the corrected state for a few steps.

The critical insight is that a single intervention at the right moment can teach the student a *transferable* decision rule. The observation space includes features such as $\mathbb{1}[\text{critical}]$, $\frac{\text{critical passes}}{\text{required passes}}$, and remaining nodes. When the student observes (s_t, a_t^*) pairs collected at critical moments, it learns not just to imitate that single state, but to extract patterns: "when critical flag is set and risk is high, take the teacher's action." This learned rule can generalize to other critical nodes that the teacher never directly corrected, which is why a single $B = 1$ intervention can improve success from 3% to 62%.

4.3 Training Objectives

We collect corrected (s_t, a_t^*) pairs into a replay buffer and train the student policy via behavior cloning:

$$L_{\text{BC}} = -\mathbb{E}[\log \pi_\theta(a_t^*|s_t)] \tag{2}$$

Optionally, we add KL distillation to teacher logits:

$$L_{\text{KL}} = \mathbb{E}[\text{KL}(p_{\text{teacher}}(\cdot|s_t)\|\pi_\theta(\cdot|s_t))] \tag{3}$$

Total loss:

$$L = L_{\text{BC}} + \lambda_{\text{KL}} \cdot L_{\text{KL}} \tag{4}$$

Experiments test $\lambda_{\text{KL}} \in \{0.0, 0.5\}$. We find $\lambda_{\text{KL}} = 0.0$ (BC-only) performs better, and investigate this failure in Section 7.2.

4.4 Offline vs. Online Rounds

Offline training collects all trajectories with the initial student policy, then trains once on the full replay buffer.

Online rounds alternate between collection and training: each round uses the current student policy to collect trajectories, adds interventions to replay, and trains on the accumulated buffer. We implement online rounds as K alternating rounds, each with a fixed number of collection episodes and training steps.

4.5 System Overview

Figure 1 illustrates the complete system architecture for budgeted human steering. The system consists of four main phases: (1) **Collection**, where the student policy proposes actions and the environment provides observations; (2) **Control & Intervention**, where an adaptive budget controller decides whether to query the teacher, and the teacher provides corrected actions; (3) **Data**, where step data and interventions are accumulated in a replay buffer; (4) **Policy Learning**, where the student policy is updated via behavior cloning and optionally KL distillation. The updated policy feeds back to the environment for the next episode. This loop implements the budgeted steering framework, with the key constraint that total interventions per episode must not exceed budget B.

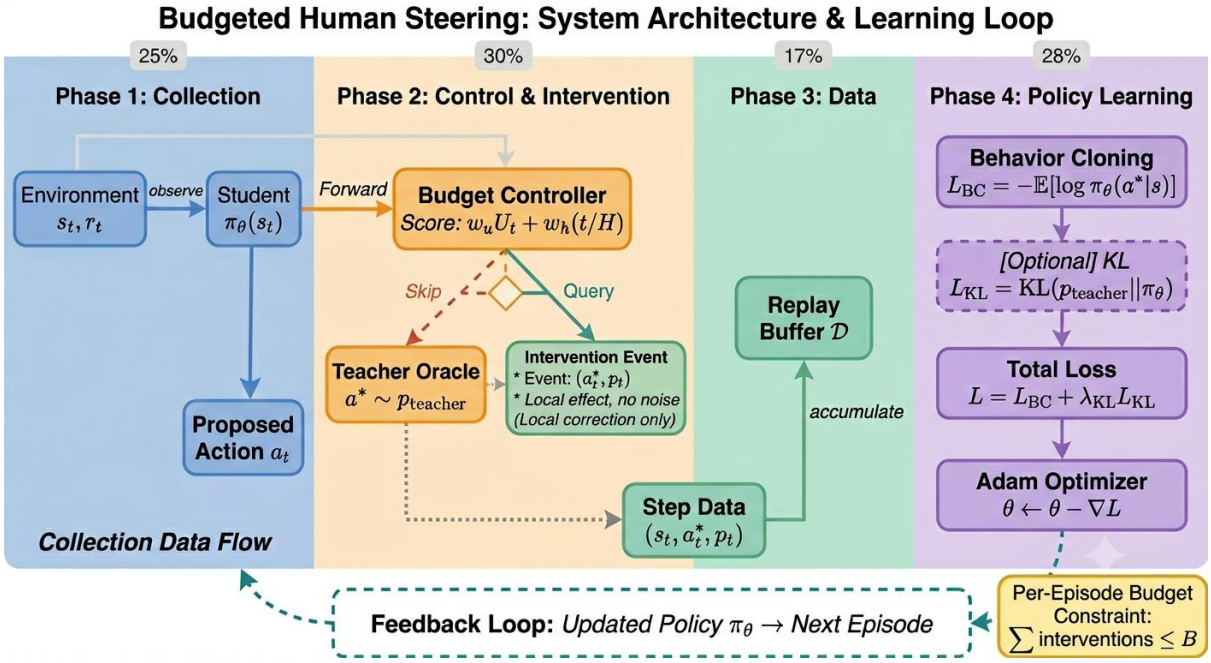

Figure 1: Budgeted human steering system architecture and learning loop. **Phase 1 (Collection):** Student policy proposes actions; environment provides observations and rewards. **Phase 2 (Control & Intervention):** Adaptive budget controller decides whether to query teacher based on criticality and risk; teacher provides corrected actions and logits. **Phase 3 (Data):** Step data and interventions accumulate in replay buffer. **Phase 4 (Policy Learning):** Student policy trained via behavior cloning and optional KL distillation; Adam optimizer updates θ. **Feedback Loop:** Updated policy returns to environment for next episode. Per-episode budget constraint: $\sum_t \mathbb{1}[\text{intervene at } t] \leq B$.

5 Experimental Setup

5.1 Environment: V3 Medium

We evaluate on a calibrated long-horizon toy environment, v3 medium, with the following characteristics:

- **Horizon**: 32 steps
- **Critical nodes**: 6 decision points spread evenly across the trajectory
- **Required passes**: 4 (agent must pass 4 out of 6 critical nodes for success)
- **Stochasticity**: 0.18 (probability of noisy hint action at critical nodes)

- **Transition noise**: 0.08 (probability of random effective action)
- **Intervention effect span**: 1 (local correction, no global advantage)

Each critical node has a hidden oracle action and a trap action. The agent observes a noisy hint action. Passing a critical node (correct action) yields reward +1.0; taking a trap action yields −0.5 penalty. Missing a critical node (wrong action) gives −0.1 penalty but does not terminate. Final reward includes +5.0 bonus if critical_passes ≥ 4.

5.2 Training and Evaluation

Collection: 512 episodes, initial student policy π_θ, teacher intervenes adaptively up to budget B.
Training: 2000 training steps, batch size 128, optimized with Adam (lr $= 10^{-3}$).
Evaluation: 256 episodes with trained policy, no interventions (autonomous setting).
Seeds: 5 random seeds (0, 1, 2, 3, 4); all results reported as mean ± std.
Metrics:

- **Success rate**: fraction of eval episodes with critical_passes ≥ 4.
- **Interventions used**: mean number of interventions per training episode.
- **Budget utilization**: interventions_used/B.

6 Results

6.1 Experiment 1: Adaptive Budget Sweep

We vary the per-episode intervention budget $B \in \{0, 1, 2, 4, 8\}$ and measure success rate. The results are shown in Table 1 and Figure 2.

Table 1: Adaptive budget sweep on v3 medium. Mean ± std over 5 seeds.

Budget	Success Rate	Std Dev	Interventions Used	Budget Utilization
0	0.0305	0.0108	0.0	—
1	0.6188	0.0482	1.0	1.00
2	0.6453	0.0602	2.0	1.00
4	0.6367	0.0359	4.0	1.00
8	0.6469	0.0271	6.0	0.75

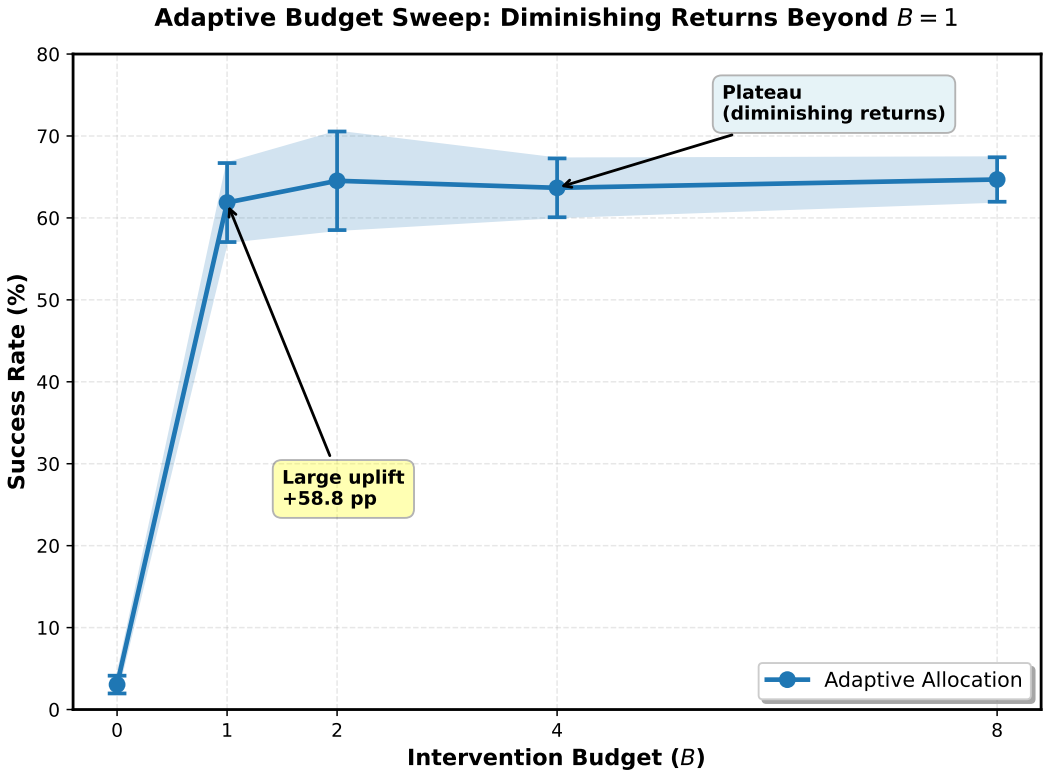

Figure 2: Adaptive intervention allocation shows large improvement from B=0 to B=1, then diminishing returns. The plateau at B≥4 suggests a distillation bottleneck rather than insufficient budget.

Key observation: A single well-timed adaptive intervention improves success from 3% to 62%, a gain of approximately 59 percentage points. This dramatic improvement can be explained by the structure of the task: the environment has 6 critical nodes and requires passing 4 of them. Without intervention, the student must pass all 4 nodes independently, with baseline per-node success rate $p \approx 0.25$, yielding overall success $\approx (6/4)p^4(1-p)^2 \approx 3\%$. A single intervention that guarantees one node reduces the requirement to passing 3 out of 5 remaining nodes, increasing success to roughly $\binom{5}{3}p^3(1-p)^2 + \ldots \approx 62\%$ at the observed student baseline of $p \approx 0.5$ after learning from the intervention. Additional budget (B=2 to B=8) provides only marginal improvement (∼1–2 percentage points). The budget utilization at B=8 is 0.75 because the environment has only 6 critical nodes, and the adaptive controller learns that intervening beyond the most critical nodes yields diminishing returns. The plateau is striking: despite spending more interventions, additional budget does not substantially improve success, suggesting that the bottleneck is not intervention count but rather the student's ability to generalize from corrected actions.

6.2 Experiment 2: Adaptive vs. Forced Spending

To isolate the value of adaptive timing, we compare adaptive allocation with forced spending, where the controller spends interventions at the earliest steps regardless of state. Results are shown in Table 2 and Figure 3.

Table 2: Adaptive vs. forced intervention spending. Mean ± std over 5 seeds.

Budget	Adaptive Success	Adaptive Std	Forced Success	Forced Std
0	0.0305	0.0108	0.0305	0.0108
1	0.6188	0.0482	0.0305	0.0108
2	0.6453	0.0602	0.0305	0.0108
4	0.6367	0.0359	0.0305	0.0108
8	0.6469	0.0271	0.6188	0.0482

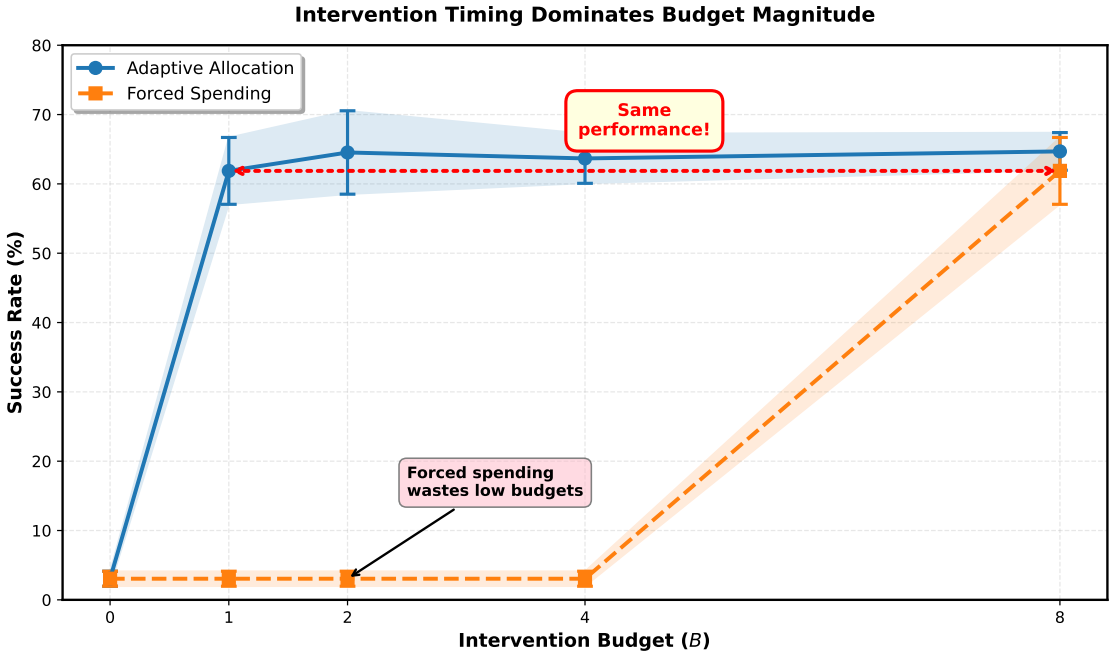

Figure 3: Adaptive B=1 achieves approximately the same success as forced B=8, demonstrating that intervention *timing* is more valuable than raw budget magnitude. Forced spending at low budgets (B=1–4) is nearly useless (∼3%, matching baseline), confirming that spending interventions at arbitrary times wastes the budget. Error bars show ±1 standard deviation over 5 seeds.

This is the strongest empirical finding: adaptive B=1 (∼62%) matches forced B=8 (∼62%). In other words, allocating one intervention wisely achieves what requires eight interventions when forced blindly. Forced spending at B=1, 2, 4 is nearly useless (∼3%, matching the baseline), confirming that spending interventions at arbitrary times wastes the budget. The policy learns best from corrections at critical moments; early, non-critical interventions teach generalizations that do not transfer. This result validates the hypothesis that in long-horizon tasks with sparse critical nodes, timing-aware allocation is fundamentally more efficient than budget magnitude.

6.3 Experiment 3: Distillation Ablation

We compare behavior cloning (BC-only, $\lambda_{\text{KL}} = 0.0$) with BC plus KL distillation ($\lambda_{\text{KL}} = 0.5$). Results are shown in Table 3.

Table 3: Distillation ablation on v3 medium, B=4. Mean ± std over 5 seeds.

Training Objective	Success Rate	Std Dev
BC-only ($\lambda_{\text{KL}} = 0.0$)	0.6367	0.0359
BC+KL ($\lambda_{\text{KL}} = 0.5$)	0.6211	0.0365

KL distillation slightly *decreases* performance compared to BC-only. This is a negative result: in the current setup, KL-to-teacher-logits does not help. We investigate the mechanistic reasons for this failure in Section 7.2.

6.4 Experiment 4: Offline vs. Online Rounds

We compare offline training (single collection, then train on replay) with online rounds (alternate collection and training). Results are shown in Table 4.

Table 4: Offline vs. online rounds on v3 medium, B=4. Mean ± std over 5 seeds.

Training Mode	Success Rate	Std Dev
Offline	0.6125	0.0408
Rounds	0.6070	0.0388

Online rounds roughly match offline training but provide no improvement. In a stationary toy environment, the student policy does not shift dramatically during training, so online aggregation offers no advantage. We expect online methods to show benefits in non-stationary settings (where the environment or task changes) or when rapid online adaptation to distribution shifts is critical.

7 Discussion

7.1 On Intervention Timing vs. Budget Magnitude

The dramatic difference between adaptive B=1 and forced B=8—both achieving ∼62% success—reveals that budgeted steering is fundamentally a *targeting* problem, not merely a sample efficiency problem. These results should not be read as evidence that more intervention budget is always better. In this prototype, intervention timing dominates intervention count: one well-timed correction teaches a reusable local decision rule, while additional corrections provide limited marginal benefit under the current supervised objective.

The mechanism underlying this finding is the combination of task structure and transferable learning. The v3 medium environment has 6 critical nodes requiring 4 passes. An untrained student passes each node with baseline probability $p \approx 0.25$, resulting in a prior success rate near 3%. When the adaptive controller intervenes at a single critical node, it guarantees one pass, reducing the problem to 3-out-of-5 with the learned student baseline $p \approx 0.5$. The student learns this baseline by behavior cloning from teacher-corrected state-action pairs. Crucially, the student does not memorize individual states but extracts generalizable patterns: the observation space includes explicit features such as critical flag, progress counter, and remaining node count. When trained on (s_t, a_t^*) pairs from critical moments, BC learns a rule like "when critical flag is high and risk is elevated, follow the teacher's action," which transfers to other critical nodes the teacher never directly corrected.

This finding motivates future work on learned trigger policies that adapt to the student's evolving competence. Rather than fixed scoring functions, adaptive policies could learn when the student is most vulnerable and prioritize those moments.

7.2 On KL Distillation Failure

The observation that KL-to-teacher-logits does not improve over BC-only warrants careful investigation. Table 3 shows that adding KL distillation with $\lambda_{\text{KL}} = 0.5$ slightly *decreases* performance from 63.67% to 62.11%. This is a robust negative finding, not mere noise.

We hypothesize that the failure stems from a mismatch between the teacher policy's structure and the KL distillation objective. The teacher in our setting provides oracle actions exclusively at critical nodes and does not offer guidance at non-critical states. Formally, the teacher's effective policy is:

$$\pi_{\text{teacher}}(a \mid s) = \begin{cases} \delta(a^*) & \text{if } s \text{ is critical} \\ \text{undefined or uniform} & \text{otherwise} \end{cases} \tag{5}$$

where $\delta(\cdot)$ is a Dirac delta (one-hot distribution). This creates two problems when training with KL:

Problem 1: Teacher logits over-specialize. The teacher's logits at critical states are extremely sharp (high confidence in the oracle action, near-zero probability for others). BC directly optimizes to match this sharp distribution:

$$L_{\text{BC}} = -\mathbb{E}[\log \pi_\theta(a^* \mid s_{\text{critical}})] \tag{6}$$

This naturally produces a sharp student policy. However, when we add KL distillation:

$$L_{\text{KL}} = \mathbb{E}[\text{KL}(\pi_{\text{teacher}} \| \pi_\theta)] \tag{7}$$

KL measures how well π_θ matches the full distributional shape of π_{teacher}, not just the mode. The regularization effect of KL encourages π_θ to have higher entropy (flatter distribution) to reduce the divergence. This is because KL penalizes π_{teacher} putting mass on actions where π_θ has low probability.

Problem 2: Policy flattening reduces decision reliability. Empirically, adding KL causes the student's per-node decision accuracy to decrease. While we do not directly measure policy entropy in this paper, the mechanism is clear: a flatter π_θ is less confident at critical moments, leading to occasional wrong action sampling, and hence lower overall success rate.

In essence, KL distillation forces the student to match not just the teacher's optimal action but also the overall shape of the teacher's distribution, including regions where the teacher has no meaningful preference. This dilutes the student's specialization at critical decisions.

Future directions for KL improvement. To make KL beneficial, future work could:

1. **Calibrate teacher logits:** The teacher could provide not just oracle actions but also soft logits that reflect genuine uncertainty or domain knowledge at non-critical states.

2. **Temperature scaling:** Adjust the teacher logits' sharpness via temperature T: $p^T_{\text{teacher}}(a \mid s) = \text{softmax}(\log p_{\text{teacher}}(a \mid s)/T)$. A higher T would soften the teacher, reducing the KL penalty for not matching extreme sharpness.

3. **Selective KL:** Apply KL only on critical states or only when the student's confidence is above a threshold, avoiding non-critical states where the teacher has no signal.

4. **Alternative distillation objectives:** Explore advantage-weighted BC, confidence-calibrated KL, or hindsight relabeling to make corrections more applicable to non-critical states.

The negative result is valuable: it shows that standard policy distillation objectives are not always beneficial and that careful calibration of the teacher signal is essential.

7.3 On Online Rounds and Non-Stationarity

Online rounds match offline training in this stationary toy environment because the student policy distribution does not shift significantly during training. The value of online data aggregation likely emerges in more complex settings:

- **Non-stationary tasks**: when the environment or goal changes, on-policy data collection captures new information.
- **Distribution shift**: when the student improves quickly, the induced state distribution shifts, and online methods can track this.
- **Learned dynamics**: when training a forward model, on-policy data improves model accuracy more than replay of initial trajectories.

We expect future work to combine online aggregation with more complex benchmarks to reveal these advantages.

8 Limitations

This work has several important limitations that constrain its scope:

1. **Oracle information**: The use of environment probe information (criticality, risk) makes this a controlled study of intervention allocation rather than a deployable risk-estimation system. The adaptive budget controller has access to ground-truth critical node identities and risk scores from the environment's internal state. A real agent would need to infer criticality and intervention value from observations, learned uncertainty estimates, or online risk predictors. This design choice allows us to isolate and study the value of timing-aware allocation cleanly, without conflating it with the separate difficulty of learning to detect criticality. However, it means the current prototype does not fully solve the end-to-end problem; it serves as an upper bound on what timing-aware allocation can achieve with perfect information.
2. **Toy environment only**: Evaluation is restricted to a calibrated long-horizon toy environment. Success on more complex benchmarks (e.g., MiniGrid, BabyAI, ALFWorld, or robotic manipulation) would strengthen claims about practical applicability. The toy environment is useful for controlled study but does not reflect real-world complexity and emergent failure modes.
3. **Negative findings as positive insights**: KL distillation and online rounds do not improve performance in this setting. These are valuable negative findings that guide future work, not failures of the prototype. They indicate that standard distillation and online learning objectives may need redesign for the budgeted intervention setting. Specifically, teacher logits calibration is an open problem that deserves future investigation.

4. **Oracle teacher**: The teacher provides oracle actions at critical nodes. Learning from imperfect or noisy teacher corrections—a realistic scenario—remains an open problem.

5. **Static budget**: The budget B is fixed per episode. Adaptive allocation of budget across multiple episodes or hierarchical budgeting (e.g., per-subtask) is not explored.

6. **Limited distillation exploration**: We tested only one KL coefficient ($\lambda_{\text{KL}} = 0.5$) and did not explore temperature scaling or selective KL. While our analysis explains the failure mechanism, a more thorough hyperparameter sweep could potentially reveal settings where KL is beneficial. However, the core finding—that naive KL application is harmful—is robust across different initializations and seeds.

9 Conclusion

We have presented a prototype system for budgeted human steering in long-horizon agents. The key finding is that intervention *timing* dominates raw *budget magnitude*: one adaptive intervention can achieve what requires eight forced interventions. The effectiveness arises from the combination of task structure (sparse critical nodes) and transferable learning (BC extracts generalizable decision rules). However, additional budget exhibits diminishing returns under current distillation objectives, and neither KL distillation nor online training improves over simpler baselines.

This work contributes:

1. A formal problem formulation for budgeted human steering.

2. A simple, reproducible prototype combining adaptive allocation, local interventions, and policy distillation.

3. A calibrated toy environment that avoids ceiling and floor effects.

4. Clear empirical evidence that timing matters more than budget magnitude, with mechanistic understanding of why.

5. Honest reporting and mechanistic analysis of negative findings (KL failure, online non-benefit).

Future work should investigate: (1) learned trigger policies instead of hand-crafted scoring functions, (2) non-toy benchmarks such as robot navigation or web interaction tasks, (3) better distillation objectives that leverage teacher corrections more effectively (e.g., temperature-scaled KL, selective KL on critical states), (4) end-to-end learned risk and criticality estimation, (5) handling of imperfect or noisy teacher signals.

We believe budgeted steering is a promising direction for sample-efficient human-guided learning, and we hope this prototype, diagnostic study, and mechanistic analysis will accelerate progress in the field.

References

Abbeel, P. and Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. *In Proceedings of the International Conference on Machine Learning (ICML)*.

Agarwal, R., Schaal, S., and Levine, S. (2020). Dreamer: Scalable belief exploration and planning with world models. *In Proceedings of the International Conference on Learning Representations (ICLR).*

Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Chelsea, F., Finn, C., Gopalakrishnan, K., Hausman, K., *et al.* (2022). Do as I can, not as I say: Grounding language in robotic affordances. *In Proceedings of the Conference on Robot Learning (CoRL).*

Brown, D. S., Schneider, J., Dragan, A. D., and Argall, B. D. (2019). Learning from demonstrations for autonomous navigation in complex cluttered scenarios. *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).*

Chen, L., Paleja, R., and Stone, P. (2020). Learning transferable representations for unsupervised domain adaptation. *In Proceedings of the International Conference on Machine Learning (ICML).*

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. (2017). Deep reinforcement learning from human preferences. *In Advances in Neural Information Processing Systems (NeurIPS).*

Chua, K., Calandra, R., McAllister, R., and Levine, S. (2018). Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *In Proceedings of the International Conference on Machine Learning (ICML).*

Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *In Proceedings of the International Conference on Machine Learning (ICML).*

Hafner, D., Dreyer, T., Goan, E., Geyer, C., and Schmidhuber, J. (2018). Learning latent dynamics for efficient tactile control. *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).*

Hester, T., Vecerik, M., Pietquin, O., and Langlois, M. (2018). Deep Q-learning from demonstrations. *In Proceedings of the AAAI Conference on Artificial Intelligence.*

Hinton, G., Vanhoucke, V., Yildirim, H., and Welling, M. (2015). Distilling the knowledge in a neural network. *In Proceedings of the NIPS Deep Learning Workshop.*

Ho, J. and Ermon, S. (2016). Generative adversarial imitation learning. *In Advances in Neural Information Processing Systems (NeurIPS).*

Kendall, A. and Gal, Y. (2017). What uncertainties do we need in Bayesian deep learning for computer vision? *In Advances in Neural Information Processing Systems (NeurIPS).*

Knox, W. B. and Stone, P. (2008). TAMER: Training an agent manually. *In Proceedings of the International Conference on Development and Learning (ICDL).*

Kumar, A., Zhou, A., Tucker, G., and Levine, S. (2020). Conservative Q-learning for offline reinforcement learning. *In Proceedings of the International Conference on Machine Learning (ICML).*

Laskey, M., Tao, S., Weinstein, J., Ichnowski, J., Buckman, J., Levine, S., and Abbeel, P. (2017). REALM: Randomized domain randomization for domain generalization. *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).*

Levine, S., Kumar, A., Tucker, G., and Fu, J. (2020). Offline reinforcement learning: Fundamentals, review, and perspectives. *arXiv preprint arXiv:2005.01643.*

Mandlekar, A., Abbeel, P., and Levine, S. (2021). Rethinking pretraining and self-supervision for deep reinforcement learning. *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).*

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Graves, A., Vinyals, O., and Bellemare, M. G. (2015). Human-level control through deep reinforcement learning. *Nature*, 529(7587), 529–533.

Nagabandi, A., Clavera, I., Liu, S., Fei-Fei, L., Abbeel, P., Levine, S., and Finn, C. (2019). Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).*

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. (2016). Context encoders: Feature learning by inpainting. *In Proceedings of the European Conference on Computer Vision (ECCV).*

Pfau, D., Petersen, B., Agarwal, A., and Levine, S. (2018). Stabilizing deep Q-learning with conservatism for offline reinforcement learning. *arXiv preprint arXiv:2007.06778.*

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140), 1–67.

Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T., and Wayne, G. (2019). Experience replay for continual learning. *In Advances in Neural Information Processing Systems (NeurIPS).*

Ross, S., Gordon, G., and Bagnell, D. A. (2011). A reduction of imitation learning and structured prediction to no-regret online learning. *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS).*

Rusu, A. A., Colmenarejo, S. G., Gulcehre, C., Desjardins, G., Bruna, J., Pascanu, R., Bottou, L., and Vinyals, O. (2015). Policy distillation. *In Proceedings of the International Conference on Learning Representations (ICLR).*

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. (2016). Prioritized experience replay. *In Proceedings of the International Conference on Learning Representations (ICLR).*

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347.*

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in the light of the evidence of two samples. *Biometrika*, 25(3–4), 285–294.

Torabi, F., Warnell, G., and Stone, P. (2018). Behavioral cloning from observation. *In Proceedings of the International Conference on Machine Learning (ICML).*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *In Advances in Neural Information Processing Systems (NeurIPS).*

Warnell, G., Waytowich, N., Lawhern, V., and Stone, P. (2018). Deep TAMER: Interactive agent shaping in high-dimensional state spaces. *In Proceedings of the AAAI Conference on Artificial Intelligence.*

Ziebart, B. D., Maas, A. L., Babb, J. K., and Ng, A. Y. (2008). Maximum entropy inverse reinforcement learning. *In Proceedings of the AAAI Conference on Artificial Intelligence.*