RELATIVE DEADLINE REGRET FOR GOAL-CONDITIONED REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Goal-conditioned reinforcement learning trains a policy to reach a desired goal from the current state with a success signal provided only upon arrival in the sparse reward setting.Policy learning near obstacles is difficult because actions that begin useful detours can receive the same immediate reward as actions that make no progress.Bellman updates compare these actions through estimated future rewards whose errors can affect policy optimization.Relative Deadline Regret (RDR) is proposed to compare candidate actions through completion probability, the probability of first arrival within a step limit called a deadline.Regret measures the completion probability lost against the best candidate as a share of the positive probability of that candidate.An arrival predictor learns from observed arrivals and unfinished attempts to compare candidate actions without Bellman action value targets.The policy fits a weighted average that favors smaller regrets across deadlines and follows a local direction of higher predicted completion called the completion gradient.Theoretical analysis bounds candidate and executed action losses and establishes convergence under stated stability conditions for the expected policy fitting update.Experimental results show the highest success rate averaged equally over the evaluated manipulation and navigation tasks and connect the candidate comparisons to learned actions near obstacles.

1 INTRODUCTION

Goal-conditioned reinforcement learning trains an agent to reach a desired goal from its current state, supporting tasks such as object manipulation and navigation (Andrychowicz et al., 2017; Park et al., 2025a). With sparse rewards, success provides direct feedback only when the goal is reached. A common approach learns action values for discounted future rewards and updates the policy toward actions with higher values. Bellman bootstrapping estimates these values from the observed reward and a value estimate at the next state (Fujimoto et al., 2018). Near obstacles, however, the policy must distinguish actions whose immediate rewards provide little evidence of their later usefulness.

An action that starts a useful detour can receive the same immediate reward as an action that makes no progress. Choosing between them requires a comparison of later outcomes. Bellman bootstrapping supplies this comparison through estimated future rewards, but errors can propagate across value updates and distort the resulting action preferences. Arrival timing adds another difficulty: a detour may be effective under a longer deadline even when completion under a short deadline is unlikely. Useful supervision should therefore compare alternative actions at several arrival deadlines and account for how attainable each deadline is.

Existing methods use goal relabeling, policy fitting and arrival prediction to make sparse outcomes more useful for policy learning. Goal relabeling changes the desired goal to an achieved goal through Hindsight Experience Replay, or HER (Andrychowicz et al., 2017; Zheng et al., 2024b). This provides training goals even when the original goal is missed. Policy fitting uses recorded actions as targets for later achieved goals through Goal-Conditioned Supervised Learning, or GCSL (Ghosh et al., 2021). Such direct policy fitting avoids Bellman action value targets. Distance Weighted Supervised Learning, or DWSL, favors recorded actions that reduce distances estimated from step count distributions (Hejna et al., 2023). Arrival prediction preserves more timing information through Survival Value Learning, or SVL (Nguimatsia Tiofack et al., 2026b). SVL learns from observed arrivals and observations ending before arrival, which leave later success unknown. Relabeling alone

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.

does not replace Bellman updates, and direct fitting emphasizes recorded actions. SVL converts arrival predictions into a discounted value, aggregating arrival times through one prescribed temporal weighting.

We propose Relative Deadline Regret (RDR) to compare the current policy action with sampled alternatives near obstacles. A completion predictor estimates first arrival within each deadline from observed arrivals and unfinished trajectories. At each deadline, RDR divides the completion probability lost against the best candidate by that candidate's positive completion probability. Figure 1 shows the policy action a_π with predicted completion 0.40 and the best candidate a_1 with 0.75 at the same deadline. The relative loss is about 0.47, so this deadline favors a_1 when assigning fitting weights. The normalization distinguishes a weak action from a deadline that is difficult for every candidate: the same absolute completion gap carries greater relative loss when the best candidate has low completion probability. RDR combines these comparisons across deadlines and fits a weighted average action, together with a local gradient of predicted completion. Candidate comparisons thus guide the policy toward alternatives that the recorded action alone would not provide as fitting targets.

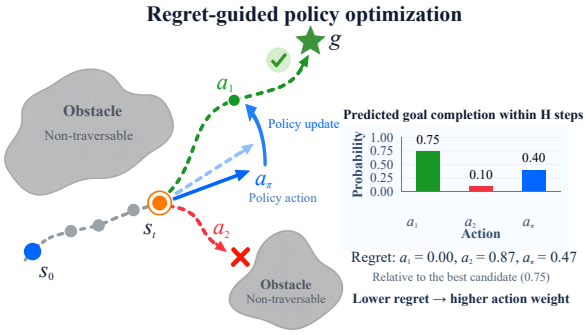

Figure 1. The illustration compares the policy action with alternative candidates by predicted completion at a shared deadline, where lower regret gives an action greater weight for policy fitting.

The contribution is a policy learning objective that uses relative completion losses across deadlines without Bellman action value targets. Theorem 1 establishes convergence of the expected policy update under a displacement stability condition, and Theorem 2 relates its limit to discounted first arrival return through explicit comparison and utility mismatch terms. Experiments on ten manipulation and navigation tasks show the highest aggregate final success among the compared methods, while ablations and spatial analyses examine how candidate fitting guides actions near obstacles.

2 Related Work

Goal-conditioned learning improves the use of sparse experience through goal relabeling and state grouping (Zheng et al., 2024b; Wang et al., 2024a), contrastive objectives (Zheng et al., 2024a; Gong et al., 2024), and exploration with intermediate goals (Bae et al., 2025; Li et al., 2025; Wang et al., 2024c). Goal relabeling and state grouping decide which transitions carry a training signal, and contrastive objectives decide how states and goals are represented. Temporal and route models describe elapsed time or extended actions (Myers et al., 2024; 2025; Ahn et al., 2025; Zheng et al., 2026). Planning methods connect trajectory segments to reach more distant goals (Bonnavaud et al., 2024; Wang et al., 2024b; Kim et al., 2024), while Merlin learns from reversed trajectories that move away from goals (Jain & Ravanbakhsh, 2024). These approaches motivate using outcomes beyond the immediate reward when a useful route requires a detour.

Time limits and arrival deadlines provide explicit temporal criteria for action selection (Pardo et al., 2018; Boyan & Littman, 2000; Cao et al., 2017; Akella et al., 2023). Each deadline states how many steps are allowed, so an action can be judged by its chance of arriving within that budget. DWSL weights recorded actions by estimated distance (Hejna et al., 2023), SVL converts first arrival predictions into discounted values (Nguimatsia Tiofack et al., 2026b), and dwell time models track continued goal attainment (Nguimatsia Tiofack et al., 2026a). More generally, policy fitting can weight scored actions (Abdolmaleki et al., 2018; Peng et al., 2019) or represent distinct action groups (Park et al., 2025b; Dong et al., 2026; Wang et al., 2025). Such weighting favors recorded actions with high scores, while a candidate comparison scores alternatives that the recorded data may not contain. RDR connects arrival prediction to candidate-based policy fitting by normalizing completion losses at each deadline before combining them into action preferences. Deadlines enter RDR as

separate comparisons rather than as one discounted value, so a deadline that favors an alternative remains visible in the fitting weight.

3 BACKGROUND

The goal-conditioned Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{G}, P, r, \mu_0, p_G)$ has state, feasible action and goal sets $\mathcal{S}, \mathcal{A}, \mathcal{G}$. An episode is one attempt starting at $s_0 \sim \mu_0$ with fixed goal $g \sim p_G(\cdot \mid s_0)$. The stationary policy $\pi(\cdot \mid s_t, g)$ assigns action probabilities by a rule independent of remaining time. The transition law $P(\cdot \mid s_t, a_t)$ gives the next state distribution and reward $r(s, a, s', g)$ indicates goal attainment. The achieved goal $\phi(s)$ is the outcome in s. Attainment $z_g(s) = \mathbf{1}[d(\phi(s), g) \le \varepsilon_g]$ uses distance d and tolerance ε_g. The indicator $z_g(s)$ is one upon attainment and zero otherwise. From $z_g(s) = 0$, first arrival time is $T_g = \inf\{j \ge 1 : z_g(s_j) = 1\}$ or infinity if never reached. The probability $F^{\pi}_H(s, a, g) = \mathbb{P}_\pi(T_g \le H \mid s_0 = s, a_0 = a, g)$ measures arrival within H steps after executing a and following π thereafter. Deadlines are integers $1 \le H \le H_{\max}$ with finite $H_{\max}$. Arrival remains counted after leaving the goal, so completion cannot decrease as H grows.

4 METHOD

RDR learns action comparisons from arrival labels and uses them for policy fitting, as Figure 2 shows. The policy network $\mu_\theta(s, g)$ outputs one action for each state and goal, and the executed policy π_θ clips this output to the allowed range through $\Pi_{\mathcal{A}}$. The behavior policy collects exploratory trajectories that supply training tuples (s_t, a_t, g, m, δ) describing arrival or nonarrival within an observation window. The predictor $f_\xi(s, a, g)$ outputs conditional arrival probabilities up to $H_{\max}$, which determine the completion probability $\widehat{F}_{\xi,H}(s, a, g)$. Policy learning fixes a state and goal pair and compares the policy action with perturbations of its first step. The predictor is trained on observed trajectory actions, while the perturbed candidates provide policy fitting targets. Relative deadline regret compares each candidate with the best candidate at the same deadline. Deadline weights combine these comparisons across step limits, and candidate weights turn them into a soft average action. The weighted average gives the fitting target, and the local completion gradient adds a direction of higher predicted completion. The next update forms new candidates from the updated policy, and execution needs only the current state and desired goal.

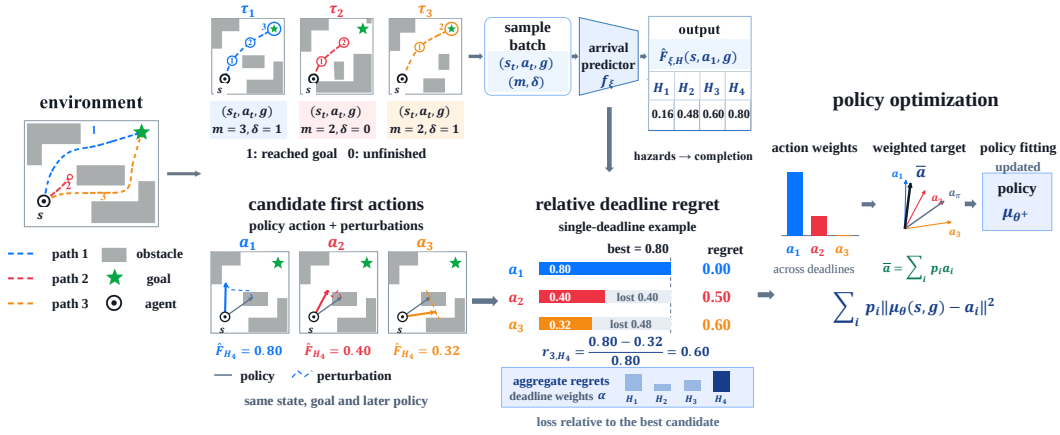

Figure 2. RDR converts recorded trajectories into completion predictions and policy fitting targets.

4.1 LEARNING COMPLETION PROBABILITIES FROM TRAJECTORIES

The behavior policy π_b collects exploratory trajectories $\tau = (s_0, a_0, \ldots, a_{L-1}, s_L)$. Segments sampled from replay use either the original desired goal or a later achieved goal through hindsight relabeling. An observed arrival specifies when the goal was reached; a censored observation specifies how long the goal remained unreached while leaving its later arrival time unknown.

Each observation starts at $0 \le t < L$ with $z_g(s_t) = 0$ and covers at most $c_t = \min\{L - t, H_{\max}\}$ steps. The time $\tau_g = \inf\{u > t : z_g(s_u) = 1\}$ is the first arrival after this start. The observation length and arrival indicator are $(m_t, \delta_t) = (\tau_g - t, 1)$ if $\tau_g - t \le c_t$ and $(m_t, \delta_t) = (c_t, 0)$ otherwise. The resulting training tuple is $(s_t, a_t, g, m_t, \delta_t)$. The predictor $f_\xi(s_t, a_t, g) = (\lambda_{\xi,1}, \ldots, \lambda_{\xi,H_{\max}})$ outputs a hazard at each step. The hazard $\lambda_{\xi,j} \in (0, 1)$ is the predicted probability of first arrival at step j given no earlier arrival. We use FHT for first hitting time, equivalently first arrival. The predictor minimizes the censored likelihood loss in Equation 1 using labels (m, δ) as in SVL (Nguimatsia Tiofack et al., 2026b).

$$\ell_{\mathrm{FHT}}(\xi; s_t, a_t, g, m, \delta) = -\delta \log \lambda_{\xi,m}(s_t, a_t, g) - \sum_{j=1}^{m-\delta} \log\big(1 - \lambda_{\xi,j}(s_t, a_t, g)\big). \tag{1}$$

An arrival label ($\delta = 1$) favors a high hazard at step m and low hazards before it. A censored label ($\delta = 0$) contributes nonarrival terms only through the observed window. The expected loss $\mathcal{L}_{\mathrm{FHT}} = \mathbb{E}_{\mathcal{D}}[\ell_{\mathrm{FHT}}]$ averages over the sampled tuple distribution $\mathcal{D}$. These observed labels train the predictor without Bellman targets for action values. The product of conditional nonarrival probabilities gives survival through H; its complement is the predicted cumulative distribution function (CDF):

$$\widehat{F}_{\xi,H}(s, a, g) = 1 - \prod_{j=1}^{H} \big(1 - \lambda_{\xi,j}(s, a, g)\big). \tag{2}$$

This completion prediction lies in $[0, 1]$ and cannot decrease as more steps are allowed. Exploration and goal relabeling can make the continuation in replay differ from the current policy; Appendix D.8 accounts for this difference in the prediction error. The resulting completion profile supplies the comparisons across deadlines in Section 4.2.

4.2 Policy Learning from Relative Deadline Regret

Completion predictions compare actions at the same state and desired goal (s, g). The K raw candidates a_i comprise the policy output and $K - 1$ Gaussian or uniform perturbations. Their executable versions are $z_i = \Pi_{\mathcal{A}}(a_i)$. The true comparison executes z_i first and fixes the current clipped policy π_k for every later action. The reference $M_H^k(s, g) = \max_i F_H^{\pi_k}(s, z_i, g)$ is the highest candidate completion probability at deadline H. Equation 3 defines relative deadline regret for selection probabilities $p_i \ge 0$ with $\sum_i p_i = 1$.

$$r_H^k(p; s, g) = 1 - \frac{\sum_i p_i F_H^{\pi_k}(s, z_i, g)}{M_H^k(s, g)}. \tag{3}$$

The comparison requires $M_H^k \ge \rho$ with $\rho = 0.01$ and describes randomly choosing one executable candidate using p. Regret is zero for the best candidate and one for a candidate with zero completion probability. The illustrative third candidate in Figure 2 loses $(0.80 - 0.32)/0.80 = 0.60$. Normalization gives the same absolute probability loss greater weight when the reference is smaller. Training scores the raw candidates a_i with the averaged predictor $f_{\bar{\xi}}$ whose parameters are averaged over updates. The predicted reference is $\widehat{M}_H = \max_i \widehat{F}_{\bar{\xi},H}(s, a_i, g)$ and the candidate losses are $\widetilde{r}_{i,H} = 1 - \widehat{F}_{\bar{\xi},H}(s, a_i, g)/(\widehat{M}_H + \varepsilon)$ with $\varepsilon > 0$ for stable division. The set $\mathcal{I}(s, g)$ retains tested deadlines with $\widehat{M}_H > \rho$ and uses $H_{\max}$ as a fallback if none passes. Equation 4 converts these losses into deadline weights α_H and candidate weights $\widetilde{p}_i$. Appendix D.3 shows that the candidate weights solve a regularized objective over the simplex. The parameter $\beta \ge 0$ controls the influence of average candidate loss on α_H and the preference b_H favors longer deadlines. The strength $\eta > 0$ makes $\widetilde{p}_i$ favor smaller losses averaged over these deadlines.

$$\alpha_H = \frac{\exp\big(b_H + \beta K^{-1} \sum_i \widetilde{r}_{i,H}\big)}{\sum_{u \in \mathcal{I}} \exp(b_u + \beta K^{-1} \sum_i \widetilde{r}_{i,u})} \qquad \widetilde{p}_i = \frac{\exp\big(-\eta \sum_{H \in \mathcal{I}} \alpha_H \widetilde{r}_{i,H}\big)}{\sum_j \exp\big(-\eta \sum_{H \in \mathcal{I}} \alpha_H \widetilde{r}_{j,H}\big)}. \tag{4}$$

Each set of weights sums to one with $\alpha_H = 0$ outside $\mathcal{I}$. Lemma 3 bounds the additional predicted regret on the returned deadline set. Lemma 4 accounts for clipping and policy fitting when comparing executed actions.

The policy fits the weighted average $\bar{a} = \sum_i \widetilde{p}_i a_i$ and follows a direction of increasing predicted completion. The current predictor f_ξ supplies this direction through the action gradient $v = \nabla_a \sum_H \alpha_H \widehat{F}_{\xi,H}(s,a,g)$ at $a = \mu_\theta(s,g)$. Dividing by the batch mean gradient norm with a positive offset gives the scaled direction $\widehat{v}$. The weight $w(s,g) \in [0,1]$ reduces the influence of pairs with weak completion predictions. Equation 5 combines candidate fitting with this local direction to train the policy. The candidates, weights and $\widehat{v}$ are held constant when differentiating the loss with respect to policy parameters.

$$\mathcal{L}_\pi(\theta) = \mathbb{E}_{\nu_w}\left[\sum_i \widetilde{p}_i \|\mu_\theta(s,g) - a_i\|^2 - \lambda_g \langle \mu_\theta(s,g), \widehat{v}\rangle\right]. \tag{5}$$

The distribution ν describes sampled state and goal pairs and $\mathbb{E}_{\nu_w}$ is its normalized weighted batch average with a positive denominator offset. The squared distance term penalizes deviations from $\bar{a}$. The inner product favors movement along $\widehat{v}$ with default strength $\lambda_g = 0.05$. The policy executes this fitted action after clipping it to the allowed range. Algorithm 1 adds an action magnitude penalty and specifies the weights and schedules.

4.3 Theoretical Analysis

We analyze the expected policy update and then relate its limiting action preferences to first arrival return. A policy function $u(x)$ gives a raw action at $x = (s,g)$ and executes $\pi_u = \Pi_{\mathcal{A}} u$. The expected batch loss has fitting strength h_u and action penalty strength $r > 0$. Its displacement g_u combines weighted candidate moves from u with the completion gradient. Appendix D.1 gives the coefficients and the assumptions behind them.

Theorem 1. Convergence of refreshed policy functions. *Let $\mathcal{A}$ be nonempty, closed and convex, and let $\mathcal{U}_R$ be all measurable raw policy functions with $\|u(x)\|_2 \le R$ on a fixed domain for some $R > 0$. Use the batch and complete predictor evaluation assumptions in Appendix D.1, which give measurable coefficients with $0 \le h_u(x) \le h_{\max} < \infty$, together with exact fitting $\mathcal{T}(u) = \frac{h_u u + g_u}{h_u + r}$ and the gradient step $\mathcal{G}_\tau(u) = (1 - 2\tau r)u + 2\tau g_u$ for $0 < \tau \le \frac{1}{2r}$ under the $L^2(\nu)$ metric. Assume the displacement bounds $\|g_u\|_\infty \le rR$ and $\|g_u - g_v\|_\infty \le L_g \|u - v\|_\infty$ for all $u, v \in \mathcal{U}_R$, with $0 \le L_g < r$. Both iterations stay in $\mathcal{U}_R$ and share a unique fixed point $u^\dagger$ that neither update changes. Every $u_0 \in \mathcal{U}_R$ satisfies*

$$\begin{aligned} ru^\dagger &= g_{u^\dagger}, \qquad \pi^\dagger = \Pi_{\mathcal{A}} u^\dagger, \\ \|\mathcal{T}^k(u_0) - u^\dagger\|_\infty &\le q_T^k \|u_0 - u^\dagger\|_\infty, \qquad q_T = \frac{h_{\max} + L_g}{h_{\max} + r} < 1, \\ \|\mathcal{G}_\tau^k(u_0) - u^\dagger\|_\infty &\le q_G^k \|u_0 - u^\dagger\|_\infty, \qquad q_G = 1 - 2\tau(r - L_g) < 1, \end{aligned} \tag{6}$$

and the executed policies converge to $\pi^\dagger$ at the same bounds.

The limit balances the displacement g_u against the action penalty ru. Convergence requires displacement sensitivity L_g to be smaller than r and allows conflicting deadline preferences. The next result uses arrival utility, the value of each arrival time, to relate the policy limit to rewards.

Theorem 2. Reward bound at the policy limit. *Let $\pi^\dagger = \Pi_{\mathcal{A}} u^\dagger$ be the executed limit in Theorem 1. Use a measurable Markov decision process that pays reward one at first arrival and zero otherwise. It has discount $0 < \gamma < 1$ and zero terminal continuation, with return V_γ^π and optimal return V_γ^*. Assume measurable action suprema and measurable maximizing policies for bounded value functions. At every state and goal before arrival, use Lemma 4 with continuation $\pi^\dagger$ and output $z_+ = \pi^\dagger(s,g)$, and assume its prediction bounds hold at all queried feasible actions. For a nonempty deadline set, let $D_H = \widehat{M}_H + \varepsilon$ and $C = \sum_{H \in \mathcal{I}} \alpha_H / D_H$ set the deadline weights $\omega_H = \alpha_H / (D_H C)$. They define the arrival utility $\psi_{s,g}(j) = \sum_{H \in \mathcal{I}} \omega_H \mathbf{1}(j \le H)$ and the score $S^\dagger(a) = \mathbb{E}_{\pi^\dagger}[\psi_{s,g}(T_g) \mid s, a, g]$ over arrival times $j \in \mathbb{N}_+ \cup \{\infty\}$, with $\psi_\gamma(j) = \gamma^{j-1}$ and both utilities zero at infinity. For the clipped candidates z_i, let the coverage gap $\kappa(s,g) = \sup_{a \in \mathcal{A}} S^\dagger(a) - \max_i S^\dagger(z_i)$ and the comparison bound $\mathfrak{B}(s,g) = \kappa(s,g) + E_\eta / C + t + c + 2e$ be uniformly bounded over all states and goals, and let $\Delta_\gamma(s,g) = \sup_j\{\psi_\gamma(j) - \psi_{s,g}(j)\} - \inf_j\{\psi_\gamma(j) - \psi_{s,g}(j)\}$ be the utility range. Then*

$$\|V_\gamma^* - V_\gamma^{\pi^\dagger}\|_\infty \le \min\left\{1, \frac{\sup_{s,g}\{\mathfrak{B}(s,g) + \Delta_\gamma(s,g)\}}{1 - \gamma}\right\}. \tag{7}$$

Equation 7 separates action comparison error from disagreement with the reward. The terms E_η/C and t cover finite candidate weighting and transfer to the executed action. Clipping contributes c and prediction error contributes $2e$. The utility range Δ_γ measures disagreement with the specified reward. Appendix D.2 proves the bound. The decomposition specifies how candidate coverage, prediction accuracy and transfer to the executed action affect the return bound. It also separates convergence of the policy update from agreement between its deadline preferences and a specified reward.

5 Experiments

The experiments ask whether these comparisons yield useful goal reaching policies and where they change policy actions near obstacles. The comparison covers HER (Andrychowicz et al., 2017), GCSL (Ghosh et al., 2021) and SVL (Nguimatsia Tiofack et al., 2026b). It also covers CRL (Eysenbach et al., 2022), QRL (Wang et al., 2023) and TMD (Myers et al., 2025). The environments and the interaction budget of RDR are given in Appendix B.

5.1 Evaluation Results

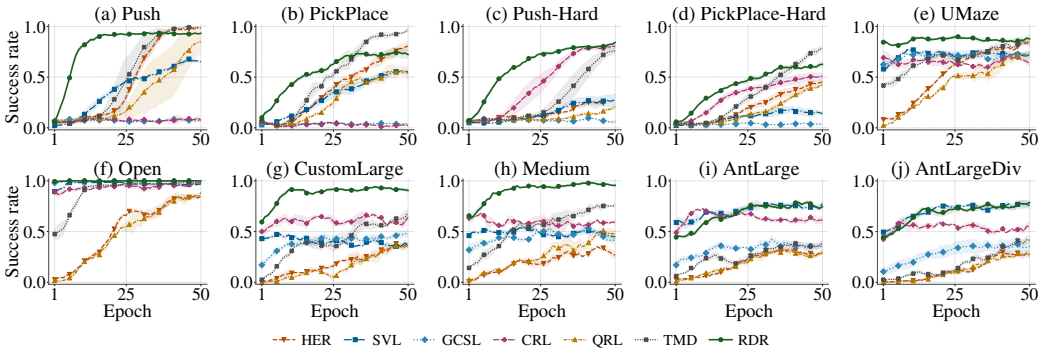

Figure 3. Learning curves show success rates for RDR and six baselines across independent runs with shaded sample standard errors, smoothed over 7 epochs.

Manipulation and maze navigation test the resulting policies through success rates during training. The short names CRL, QRL and TMD denote Contrastive Reinforcement Learning, Quasimetric Reinforcement Learning and Temporal Metric Distillation. Figure 3 plots success rate against training epoch with Fetch manipulation tasks in panels a to d and PointMaze navigation tasks in panels e to j. Shaded bands mark sample standard errors across independent runs and every curve is smoothed over 7 epochs. Across all ten tasks in Table 2, RDR has the highest equally weighted mean final success rate, 83.63%, compared with 74.75% for the strongest aggregate baseline, TMD. RDR has the highest mean on five tasks and ties for the highest mean on Open. Its advantage is clearest on tasks that require navigating around obstacles: Push Hard reaches 82.6% against the best baseline at 80.4%, CustomLarge reaches 91.1% against 64.7%, and Medium reaches 95.6% against 75.3%. The learning curves also show an earlier rise on Push and a sustained advantage through much of training on Push Hard. Together, these results show strong aggregate performance across manipulation and navigation, with particularly large gains on CustomLarge and Medium. The following analyses examine how the candidate comparisons relate to these learned behaviors.

5.2 Candidate Comparisons near Obstacles

Relative deadline regret compares actions at several arrival limits to account for the extra steps needed by a detour. The best candidate prediction $\widehat{M}_H$ supplies a reference at each deadline. The plotted relative loss $1 - \widehat{F}_H(s, \mu(s,g), g)/\widehat{M}_H$ measures the fraction of that predicted completion lost by the policy action. The diagnostic uses relative loss without the numerical offset ε. Deadline weights combine these comparisons before candidate weights define the fitting target $\bar{a}$. The distance $\|\bar{a} - \mu(s,g)\|$ then measures the action change requested by fitting. The controls $\mathrm{RDR}_{\mathrm{NoReg}}$

and $\mathrm{RDR}_{\mathrm{NoGrad}}$ remove candidate fitting and the completion gradient, respectively. The control $\mathrm{RDR}_{\mathrm{NoNorm}}$ removes probability division from both deadline and candidate weight calculations. Fixed Push Hard states reveal how these changes affect the comparisons near an obstacle, where obstructed states have an obstacle on the straight line from the object to the desired goal.

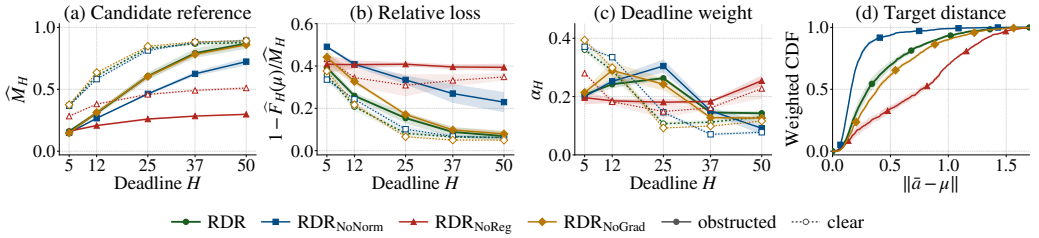

Figure 4. Push Hard diagnostics follow the comparison from predicted candidate completion to policy fitting targets, with solid curves for obstructed states and dotted curves for clear states.

Figure 4 connects the predicted probabilities to the action changes requested by fitting. Panel a shows low candidate completion at short deadlines for obstructed states and larger predictions when more arrival time is allowed. Panel b shows that the relative loss of RDR falls toward the clear state level as the deadline grows. The policy action therefore approaches the best candidate prediction in relative terms even where the predicted arrival requires more steps. The $\mathrm{RDR}_{\mathrm{NoReg}}$ loss remains larger, while $\mathrm{RDR}_{\mathrm{NoGrad}}$ follows a trend close to RDR. Panel c shows that RDR places its greatest weight on an intermediate deadline at obstructed states. The fitting target thus reflects intermediate arrival requirements as well as the longest limit. Panel d weights obstructed states by $w(s, g)$ and gives the weighted fraction below each target distance, where a curve farther left means smaller requested changes. The $\mathrm{RDR}_{\mathrm{NoNorm}}$ targets are closer to the policy action than RDR targets, while $\mathrm{RDR}_{\mathrm{NoReg}}$ targets are farther away. The $\mathrm{RDR}_{\mathrm{NoReg}}$ target is a diagnostic candidate average rather than a trained quantity. Probability division expresses small absolute gaps as relative losses when the reference prediction is small. These losses can separate candidate weights at low completion levels, consistent with the larger RDR target distances in this joint weight comparison.

5.3 Policy Actions near Obstacles

Candidate fitting turns comparisons among alternative actions into a policy output that must move around walls. The CustomLarge maps examine this connection across positions where the path to the goal requires changes in direction. The arrival budget is the first deadline at which predicted completion of the policy action reaches 0.5. Relative loss at $H = 300$ compares the policy action with its best candidate at the longest deadline. The corresponding deadline weight shows how strongly this arrival limit contributes to fitting. RDR and $\mathrm{RDR}_{\mathrm{NoReg}}$ are compared at the default gradient strength using their own predictors at fixed positions with zero velocity, and Appendix C.4 gives the query settings and the masking rule.

Figure 5 presents CustomLarge maps with shared color scales per column for RDR in the upper row and $\mathrm{RDR}_{\mathrm{NoReg}}$ in the lower row. Panel a assigns longer arrival budgets near the left side of the central wall while panel e leaves broad $\mathrm{RDR}_{\mathrm{NoReg}}$ regions below the arrival threshold. Panels b and f show the relative loss at $H = 300$ where the RDR losses stay low across the longer budget region. The larger $\mathrm{RDR}_{\mathrm{NoReg}}$ losses sit above the right passage within the smaller region it admits. Panels c and g give greater weight to the longest deadline near the edges of the predicted completion regions. Panels d and h enlarge the dashed window where red and blue mark leftward and rightward components relative to the goal direction while the $\mathrm{RDR}_{\mathrm{NoReg}}$ actions remain nearly zero. Sideways actions of RDR near the wall agree with the loss balance in Appendix D.3 where a nonzero fitting target can sustain action magnitude.

5.4 Supervision Through Training

The predictor and policy change during training, so useful comparisons must remain available as their outputs evolve. Relative loss measures how closely the policy action follows the best candidate prediction. The admitted fraction measures how widely such a comparison is available among fixed

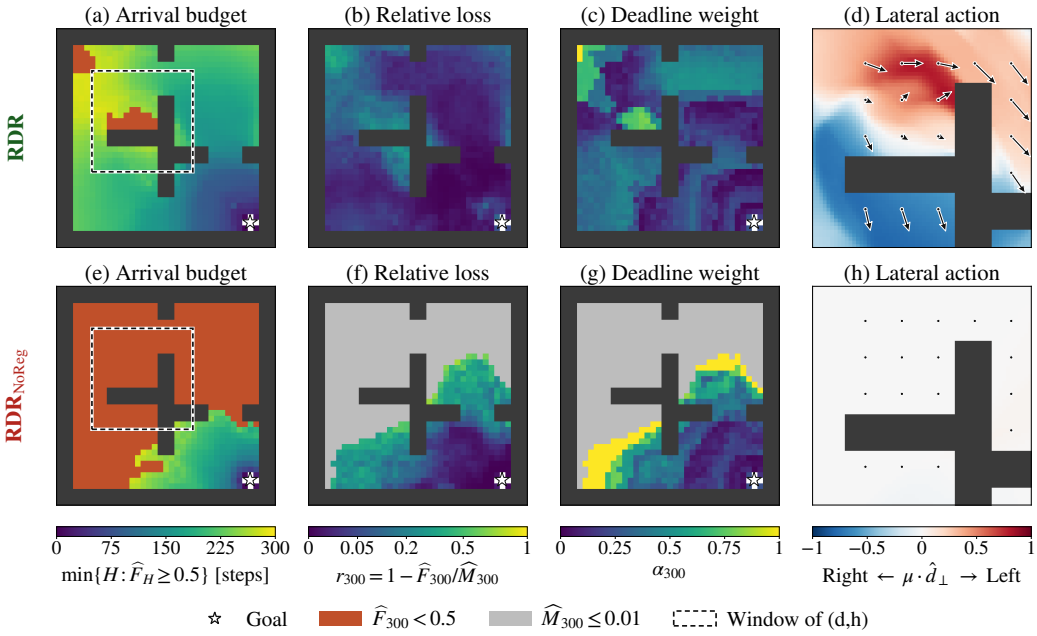

Figure 5. CustomLarge maps compare completion predictions and policy actions for RDR above and $\mathrm{RDR}_{\mathrm{NoReg}}$ below on shared column scales, where orange marks positions whose predicted completion stays below one half through $H = 300$ and grey marks positions without a usable comparison between the two actions.

queried states. States are admitted when $\widehat{M}_{H_{\max}} > \rho = 0.01$, and loss at the longest deadline is summarized over admitted states with each variant using its own subset. Tracking both quantities on Push Hard and CustomLarge distinguishes smaller relative losses from losses measured over fewer states.

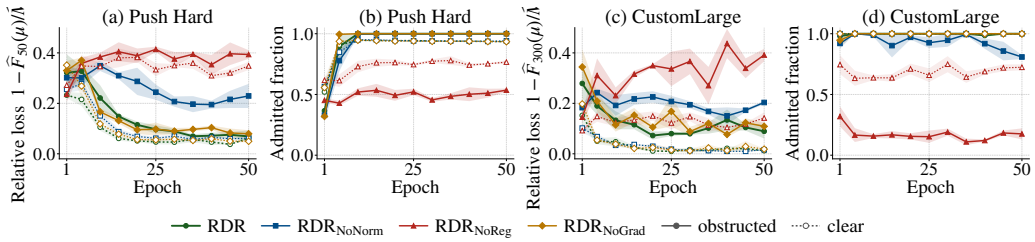

Figure 6. Training curves compare relative loss and comparison admission at obstructed and clear states on Push Hard and CustomLarge, where solid curves represent obstructed states and dotted curves represent clear states.

Figure 6 tracks relative loss and admitted fraction through training on Push Hard in panels a and b and on CustomLarge in panels c and d. In every panel the solid curves represent obstructed states and the dotted curves represent clear states. Panels a and b show the Push Hard trends where the RDR loss falls at obstructed states while the $\mathrm{RDR}_{\mathrm{NoReg}}$ losses remain larger. Panel b shows that RDR and $\mathrm{RDR}_{\mathrm{NoGrad}}$ admit more of these obstructed states than $\mathrm{RDR}_{\mathrm{NoReg}}$ later in training. Panel c shows a more variable RDR loss on CustomLarge that remains higher at obstructed than clear states. Panel d shows broad comparison admission for RDR and $\mathrm{RDR}_{\mathrm{NoGrad}}$ while $\mathrm{RDR}_{\mathrm{NoReg}}$ admits few obstructed states late in training. These trends show that comparisons remain available at the queried states for RDR and $\mathrm{RDR}_{\mathrm{NoGrad}}$ as the predictor and policy change.

5.5 Ablation Studies

RDR combines candidate fitting with a local completion gradient and probability normalization. The component controls test their roles on two obstacle manipulation tasks and two maze layouts,

keeping the retained components at their default settings. Table 3 also varies gradient strength and the deadline and action scores.

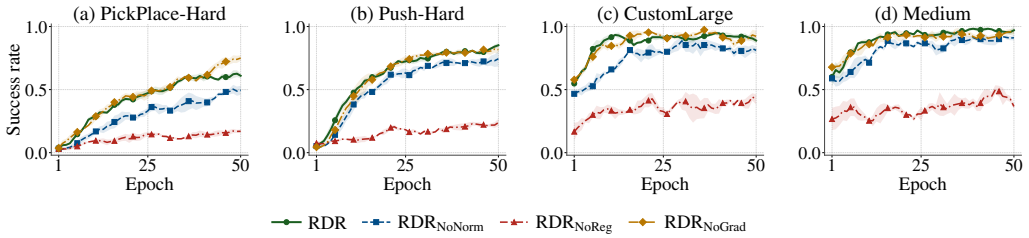

Figure 7. Component controls compare success during training on four tasks with shaded sample standard errors across independent runs, smoothed over 5 epochs.

Figure 7 shows success during training for the component controls on four tasks in panels a to d with shaded sample standard errors across independent runs, smoothed over 5 epochs. The control $\mathrm{RDR}_{\mathrm{NoReg}}$ removes candidate fitting while the completion gradient and the action penalty remain active. The control $\mathrm{RDR}_{\mathrm{NoGrad}}$ removes the completion gradient while candidate fitting and the action penalty remain active. The control $\mathrm{RDR}_{\mathrm{NoNorm}}$ removes probability division from both weight calculations, so it tests their combined change. The control $\mathrm{RDR}_{\mathrm{Fixed}}$ sets $\beta = 0$ and keeps the fixed deadline preference while its weights ignore average candidate loss. The control $\mathrm{RDR}_{\mathrm{Return}}$ replaces the completion score with one discounted arrival score at the tested $\gamma = 0.98$. The largest drops in panels a to d follow the removal of candidate fitting, where RDR reaches 82.6 on Push Hard while $\mathrm{RDR}_{\mathrm{NoReg}}$ at $\lambda_g = 0.05$ reaches 21.8. Removing probability normalization lowers the mean on all four tasks, including a decrease from 82.6 to 73.0 on Push Hard. Replacing the completion profile with the discounted score also lowers all four means, supporting comparisons across deadlines at the tested $\gamma = 0.98$. Candidate fitting alone retains much of the performance: $\mathrm{RDR}_{\mathrm{NoGrad}}$ is close to RDR on three tasks and reaches 74.2 on PickPlace Hard compared with 62.2 for RDR. The fixed deadline control is also close to the default, whereas a stronger completion gradient substantially improves $\mathrm{RDR}_{\mathrm{NoReg}}$. The clearest component evidence therefore supports candidate fitting and probability normalization under the default update; the local gradient and adaptive deadline weighting have smaller, task-dependent effects.

Taken together, the analyses describe a consistent path from arrival observations to policy actions near obstacles. The candidate comparison keeps the policy action close to the best candidate prediction as the deadline grows, and the admitted fraction stays high for RDR through training. At fixed maze positions these comparisons request sideways actions near walls where the direct path is blocked. The relative losses stay lower for RDR than for the control that removes candidate fitting, which agrees with the larger component effects in the ablations. The evidence is strongest where an obstacle forces a choice among actions, and it is weaker on the simpler Fetch tasks where several compared methods reach similar final success rates.

6 Conclusion

RDR uses arrival timing to guide goal-conditioned policy learning near obstacles. Relative completion losses compare candidate actions at several deadlines, and their weighted target guides policy fitting together with a local completion gradient. Across ten tasks, RDR achieves the highest aggregate final success among the compared methods; ablations and spatial analyses identify candidate fitting and probability normalization as its main empirical strengths. The theory characterizes a stable limit of the expected policy update and separates candidate, prediction and execution errors in its relation to first arrival return. This approach makes completion across deadlines available during learning while execution uses only the current state and desired goal.

Limitations. The benefits of the completion gradient and adaptive deadline weights vary across tasks, with some simpler variants matching or exceeding RDR. Replay-to-policy mismatch and action clipping can weaken the transfer of predicted candidate advantages to execution, and direct target substitution did not improve arrival rates at the tested states. The convergence result assumes stable expected updates, while the return bound is noninformative at the tested discount $\gamma = 0.98$.

AI USE STATEMENT

Claude by Anthropic through Claude Code and ChatGPT by OpenAI assisted with manuscript revision, literature and reference checks, figure interpretation, and review of mathematical statements and proofs. They also provided feedback on the method and experiments and assisted with consistency checks of results, LaTeX formatting and figure preparation. The authors are responsible for the final text, claims, results and artifacts.

REPRODUCIBILITY STATEMENT

Algorithm 1 specifies the training objective and goal relabeling. Table 1 lists the interaction budget of RDR for each task, and Figure 8 shows the environments. Appendix C defines the controls and diagnostic analyses. Appendix D states the assumptions and proves the theoretical results.