SHALLOW NETWORKS, DISTANT GOALS: COMPETENCE-AWARE LOCAL PLANNING FOR ONLINE GCRL

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning to reach distant goals is a central challenge in online goal-conditioned reinforcement learning (GCRL). Recent work shows that scaling the network depth of contrastive reinforcement learning (CRL) improves long-horizon goal reaching through richer representations, but at a substantial computational cost. We ask how far shallow networks can go when given a suitable planning structure. To this end, we introduce Competence-Aware Local Planning (CALP), a simple approach that builds long-horizon behavior from repeated local decisions. At each decision, CALP proposes a small set of candidate subgoals around the achieved goal, using a learned execution competence to identify subgoals the controller can actually reach and a lower-quantile temporal-distance estimate to steer selection toward the final goal. A single shared controller executes the selected subgoals, and all components are learned jointly from online interaction. Across eight navigation, locomotion, and manipulation tasks in JaxGCRL, CALP uses two-hidden-layer networks and a single local selection step per decision, yet achieves the best performance in six tasks and matches deep CRL variants of up to 64 layers in the remaining two, while training up to five times faster. These results show that local goal-space planning is an effective and efficient alternative to scaling network depth for long-horizon goal reaching.

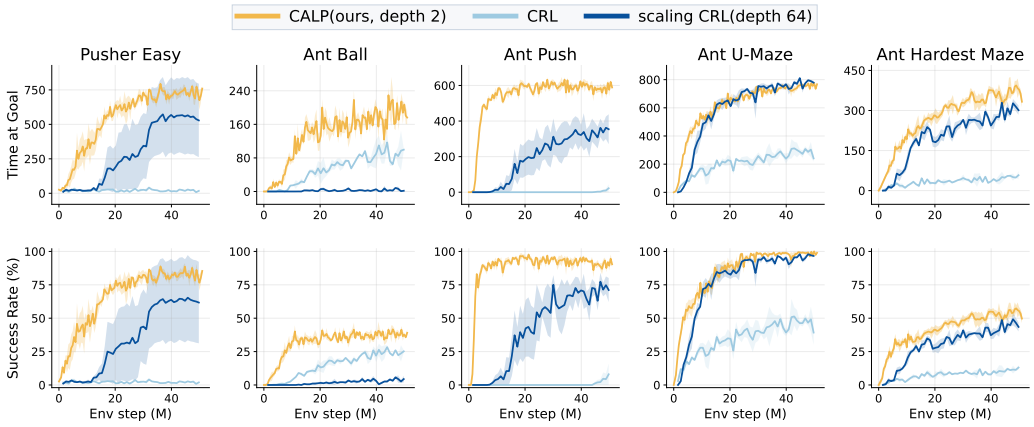


Figure 1: Time at goal (top) and success rate (bottom) of CALP (ours, 2 layers), CRL (4 layers), and Scaling-CRL (64 layers) on five goal-conditioned manipulation, locomotion, and navigation tasks. Curves show the mean over five random seeds, and shaded regions show ± one standard deviation. CALP's shallow network with planning matches or outperforms the deeper baselines.

1 INTRODUCTION

Learning to reach distant goals is a central and long-standing challenge in online goal-conditioned reinforcement learning (GCRL). Building on early work on learning to achieve goals (Kaelbling, 1993), universal value function approximators (Schaul et al., 2015) and hindsight relabeling (Andrychowicz et al., 2017), GCRL has offered a self-supervised route to acquiring broadly use-

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.

ful behaviors without hand-designed reward functions (Liu et al., 2022). Recent JaxGCRL benchmark (Bortkiewicz et al., 2025) has accelerated this line of work, training ten million environment steps in roughly ten minutes on a single GPU. Yet reaching temporally distant goals remains difficult: rewards are sparse, credit must be assigned over long horizons, and the agent must both explore and stitch experience into long-range behavior.

Contrastive reinforcement learning (CRL) has emerged as a particularly promising family of self-supervised GCRL methods. By casting representation learning as the RL problem itself, CRL learns a goal-conditioned value function directly from action-labeled trajectories (Eysenbach et al., 2022), building on recursive-classification (Eysenbach et al., 2021) and stabilization techniques (Zheng et al., 2024a). Recent work (Wang et al., 2025) showed that scaling network depth of CRL, up to 64 layers, far beyond the 2–5 layer networks typical of prior RL (Eysenbach et al., 2022; Nauman et al., 2024), can significantly boost goal-reaching capability. This mirrors a broader trend in RL, where scaling network capacity has driven gains in continuous control and Atari (Nauman et al., 2024; Lee et al., 2025; Schwarzer et al., 2023). However, this depth comes at a high computational and wall-clock cost (Figure 2): a 64-layer network takes much longer to train than its shallow counterpart, so a fixed compute or time budget yields far fewer effective training steps at greater depths.

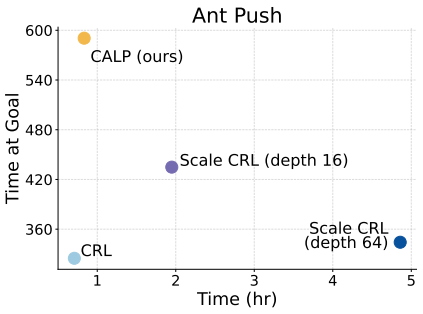


Figure 2: Performance vs. Wall-clock cost on Ant Push. Wall-clock time is averaged over five seeds (50M training steps each) on an NVIDIA RTX 4090 GPU.

Depth helps through two channels (Wang et al., 2025): better exploration and richer representations that capture long-range connectivity. This raises a natural question: *can shallow networks obtain these benefits through structure rather than capacity?* Recent findings show that shorter effective horizons scale offline goal-conditioned RL by limiting value bootstrapping errors (Park et al., 2025). We extend this idea to the online setting. We hypothesize that when an agent reliably reaches well-chosen nearby goals, repeating this process shortens the effective horizon and expands data coverage. Together, these effects enable shallow networks with suitable planning structures to learn goal-space structures that would otherwise require much deeper ones.

We introduce **Competence-Aware Local Planning (CALP)**, a simple approach that builds long-horizon behavior from repeated local decisions. At each decision point, CALP generates candidate subgoals $w = ag + \sigma_g \odot \delta_k(\pm e_i)$ around the achieved goal ag, where e_i is the i-th standard basis vector of the d_g-dimensional goal space and σ_g, the running per-dimension standard deviation of achieved goals, normalizes the step size δ_k. It then selects among these candidates using two learned quantities.

The first is execution competence, a learned estimate of which nearby subgoals the controller can actually reach. Unlike prior feasibility estimators, which are typically binary (Section 2), it regresses a graded hindsight target: the fraction of a commanded displacement that was actually realized. The second is a lower-quantile estimate of temporal distance, which steers selection toward the final goal. Fitting a lower quantile discounts the inflated targets that hindsight relabeling produces from wandering, exploratory trajectories; the resulting estimate is deliberately optimistic, and the competence filter guards against acting on it where the controller cannot follow. A single shared controller executes each selected subgoal. It is trained on an internal progress reward toward the active waypoint, computed from achieved goals alone, and its temporal-difference targets are truncated at waypoint boundaries, so credit is never assigned across different planner commands. All components are learned jointly from online interaction.

Our primary contribution is to shed light on whether shallow networks, equipped with an appropriate local planning structure, can reach the goal-reaching performance that recent work instead obtains by scaling network depth. Our results answer this question affirmatively. Across eight navigation, locomotion, and manipulation tasks in JaxGCRL (Bortkiewicz et al., 2025), CALP with two-hidden-layer actor and critic networks outperforms 16- and 64-layer CRL agents on six tasks and matches them on the remaining two, while training about five times faster. Moreover, increasing the depth of CALP's networks yields only modest improvements (Section 5.3), indicating that CALP does not rely on increasingly deep networks to achieve strong long-horizon goal-reaching performance.

2 Related Work

Goal-conditioned and contrastive RL. Goal-conditioned RL dates to Kaelbling (1993), scaled to function approximation via Schaul et al. (2015); Borsa et al. (2019), and made practical under sparse reward by hindsight relabeling (Andrychowicz et al., 2017; Liu et al., 2022). A self-supervised line recasts goal reaching as recursive classification (Eysenbach et al., 2021) or contrastive representation learning (Eysenbach et al., 2022), later stabilized (Zheng et al., 2024a) and refined for credit assignment (Zheng et al., 2024b); this family supplies our CRL baseline and, via JaxGCRL (Bortkiewicz et al., 2025), our benchmark substrate. Specifically, scaling contrastive RL to hundreds of layers unlocks new goal-reaching capability (Wang et al., 2025), while a complementary view holds that reducing the effective horizon is what actually drives scalability (Park et al., 2025). CALP follows the latter view: it builds on hindsight relabeling and an explicit achieved-goal space, and shows that decomposing long-horizon behavior into local decisions lets two-hidden-layer networks match or exceed deep contrastive agents.

Subgoal selection. Hierarchical methods train a high-level policy to propose subgoals for a low-level policy (Vezhnevets et al., 2017; Nachum et al., 2018; Levy et al., 2019; Hafner et al., 2022; Park et al., 2023), while planning-based methods search a graph of past states under a learned distance (Eysenbach et al., 2019; Nasiriany et al., 2019; Hoang et al., 2021; Beker et al., 2022). Closest to our setting, DCP and its variant SSGC (S K et al., 2026) select subgoals online from recently visited states by contrastive similarity to the final goal. Other methods choose goals to drive exploration (Pitis et al., 2020; Hu et al., 2023). CALP requires neither a high-level policy nor graph search, it generates candidates as local displacements of the achieved goal and makes a single selection per decision at constant cost.

Subgoal feasibility. Restricting subgoals to what the controller can reach is a recurring idea. HRAC (Zhang et al., 2020) constrains subgoals to a k-step adjacent region using an adjacency embedding learned from a binary reachability matrix, and kSubS (Czechowski et al., 2021) keeps only generated subgoals that a low-level policy reaches in a known environment model. Other approaches train binary reachability classifiers (Qian et al., 2023) or, as in the graph planners above, threshold a learned distance. CALP's competence model instead regresses the fraction of a *commanded* displacement actually realized, learned online from the controller's own commands. To our knowledge, no prior feasibility estimator uses such a graded, command-conditioned target.

Temporal distance estimation. Dynamical Distance Learning (Hartikainen et al., 2020) regresses hindsight elapsed time with a squared loss, estimating the mean traversal time under the data-collecting policy. Several methods instead target near-optimal values or distances from suboptimal data: IQL (Kostrikov et al., 2022) and HIQL use expectile regression within temporal-difference backups, QRL (Wang et al., 2023) recovers shortest paths through a quasimetric objective, related architectures build goal-reaching geometry into the model (Liu et al., 2023; Myers et al., 2024), and SoRB learns distances with a distributional critic. CALP pursues the same goal with a simpler estimator: a pinball loss (Koenker & Bassett, 1978; Dabney et al., 2018) fits a lower quantile of hindsight elapsed time directly, with a plain MLP and no Bellman backup or architectural constraint.

3 Preliminaries

We introduce the notation and learning principles underlying CALP: goal-conditioned reinforcement learning, hindsight supervision, deterministic actor–critic learning, and temporal distance estimation. We focus on the online setting, where the agent continually collects trajectories in a replay buffer $\mathcal{D}$ and learns control, temporal distance, and execution competence from its own experience.

Goal-conditioned RL. Goal-conditioned reinforcement learning learns a policy $\pi(a \mid s, g)$ that reaches goals $g \in \mathcal{G}$ across a shared environment (Schaul et al., 2015). Let $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $P(s' \mid s, a)$ the transition dynamics, and $h : \mathcal{S} \to \mathcal{G}$ the achieved-goal mapping. Given a goal distribution p_g and a goal-dependent reward r, the objective is

$$J(\pi) = \mathbb{E}_{g \sim p_g,\, \pi, P}\left[\sum_{t=0}^{T-1} \gamma^t r(s_t, a_t, s_{t+1}, g)\right], \tag{1}$$

where $\gamma \in [0, 1)$ is the discount factor and T is the episode horizon. Goal-reaching success is defined by a task-specific tolerance, $\|h(s_t) - g\| \leq \epsilon_g$. Throughout, g denotes the final task goal and w denotes a local subgoal selected by the planner.

Hindsight experience replay. Hindsight experience replay (HER) (Andrychowicz et al., 2017) extracts additional supervision by relabeling transitions with goals achieved later in the same trajectory. For a future index $j > t$,

$$\tilde{g} = h(s_j), \qquad \tilde{r}_t = r(s_t, a_t, s_{t+1}, \tilde{g}). \tag{2}$$

The resulting transition $(s_t, a_t, \tilde{r}_t, s_{t+1}, \tilde{g})$ supports off-policy learning from achieved outcomes. This hindsight principle also provides temporal-distance labels: the pair $(h(s_t), h(s_j))$ is associated with elapsed time $j - t$. CALP uses these pairs for distance learning and the outcomes of executed commands for competence learning.

Deep deterministic policy gradient. Deep deterministic policy gradient (DDPG) (Silver et al., 2014; Lillicrap et al., 2015) learns a deterministic actor $\mu_\phi(s, g)$ and an action-value critic $Q_\theta(s, a, g)$. The critic minimizes a temporal-difference loss, while the actor maximizes the critic's predicted value:

$$\mathcal{L}_Q(\theta) = \mathbb{E}_{\mathcal{D}} \left[(Q_\theta(s, a, g) - y)^2 \right], \tag{3}$$

$$y = r + \gamma b \, Q_{\bar{\theta}}(s', \mu_{\bar{\phi}}(s', g), g), \tag{4}$$

$$\mathcal{L}_\pi(\phi) = -\mathbb{E}_{(s,g)\sim\mathcal{D}} \left[Q_\theta(s, \mu_\phi(s, g), g) \right], \tag{5}$$

where b is the bootstrap indicator and barred parameters denote target networks. Twin Delayed DDPG (TD3) (Fujimoto et al., 2018) introduces twin critics, delayed actor updates, and target policy smoothing:

$$y_{\mathrm{TD3}} = r + \gamma b \min_{i=1,2} Q_{\bar{\theta}_i}(s', \tilde{a}', g), \qquad \tilde{a}' = \mathrm{clip}_{\mathcal{A}}\big(\mu_{\bar{\phi}}(s', g) + \epsilon\big), \tag{6}$$

where ϵ is clipped Gaussian noise. The actor is optimized through Q_{θ_1}, and target networks are updated by Polyak averaging. CALP uses TD3 to train its shared local controller.

Temporal distance and quantile regression. Temporal distance measures goal separation through the transitions required to connect states and provides a useful signal for planning (Eysenbach et al., 2019; Wang et al., 2023). For achieved goals $x_t = h(s_t)$ and $x_j = h(s_j)$ from the same trajectory, CALP uses the transformed elapsed time $Y_{t,j} = \log(1 + j - t)$ as supervision. Quantile regression (Koenker & Bassett, 1978) fits D_ψ using

$$\mathcal{L}_{\mathrm{dist}}(\psi) = \mathbb{E}_{(t,j)\sim\mathcal{D}} \left[\rho_\tau\big(Y_{t,j} - D_\psi(x_t, x_j)\big) \right], \qquad \rho_\tau(u) = u\big(\tau - \mathbb{I}\{u < 0\}\big). \tag{7}$$

This objective estimates a conditional τ-quantile of the transformed elapsed time under the replay distribution. Choosing $\tau < 1/2$ emphasizes faster observed transitions, whereas squared-error regression estimates the conditional mean. CALP combines this temporal signal with execution competence to guide local subgoal selection.

4 Long-Horizon Goal Reaching through Local Planning

This section first identifies why value-guided goal reaching fails in obstacle-rich environments, and then introduces CALP, which addresses this failure by separating local feasibility from progress toward the goal.

4.1 When Goal Guidance and Local Feasibility Diverge

Goal-conditioned actor–critic methods select actions by following a learned value function, using hindsight relabeling to obtain supervision from achieved outcomes (Schaul et al., 2015; Andrychowicz et al., 2017; Liu et al., 2022). Figure 3 visualizes this guidance for TD3+HER. Red Arrows show learned value-gradient directions, and green curves show short executed trajectories. In an open pushing configuration (Figure 3(a)), moving the object toward its target both yields immediate progress and advances the task, so the value gradients lead to executable motion and success steadily improves. In obstacle-rich navigation (Figure 3(b)), this alignment breaks down. A goal across a wall may be geometrically close yet require a long detour, and the correct initial motion may temporarily increase the Euclidean distance to it. The learned value gradients

nonetheless point toward the goal, driving the agent repeatedly into the obstacle, and success remains low.

This failure reflects a structural burden: a single value function must judge at once which motions the current policy can execute and which motions lead toward the goal, and near obstacles the two judgments conflict. Contrastive RL improves this function by learning representations that reflect dynamical connectivity rather than geometric proximity (Eysenbach et al., 2022), and scaling network depth improves it further at additional computational cost. Both approaches, however, still ask one function to resolve long-range structure, which is precisely where data is sparse. CALP instead separates the two judgments. Feasibility is estimated only for nearby commands, where transitions are densely observed and can be learned reliably, and progress is estimated only well enough to rank a few feasible candidates. The next subsection develops this idea into a simple local planning structure.

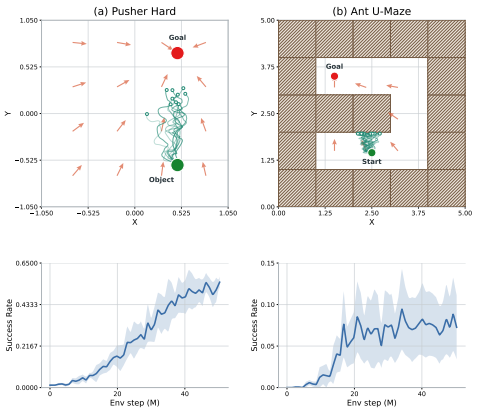


Figure 3: Goal-directed guidance and local feasibility for TD3+HER.

4.2 Competence-Aware Local Goal Planning

We now describe the components of Competence-Aware Local Goal Planning(CALP). At each decision, CALP constructs a set of nearby candidate subgoals, filters them by predicted execution competence, and selects the remaining candidate with the smallest estimated temporal distance to the final goal; a single shared controller then executes the selected subgoal (Figure 4). Each component is ablated in Section 5.3.

Local candidates. Let $z = ag(s) \in \mathbb{R}^{d_g}$ denote the achieved goal at state s, and let $\sigma_g \in \mathbb{R}^{d_g}$ be the running per-dimension standard deviation of achieved goals, which normalizes the scale of each goal coordinate. We treat the goal space as locally Euclidean, so that small displacements around z can be expressed along its coordinate axes. We therefore construct candidates using the signed basis directions $\mathcal{U} = \{\pm e_i\}_{i=1}^{d_g}$ and a finite set of normalized step sizes $\{\delta_k\}_{k=1}^{K}$:

$$\mathcal{W}(s) = \{z + \sigma_g \odot (\delta_k u) \ : \ u \in \mathcal{U},\ k = 1, \dots, K\}, \tag{8}$$

where $\odot$ denotes elementwise multiplication. This yields $2d_gK$ candidates spanning multiple directions and scales around the current achieved goal, each specifying a local displacement for the shared controller.

Learning execution competence. We learn an execution-competence model $C_\phi(s, \delta) \in \mathbb{R}^{2d_g}$ that, given the current state s and a commanded normalized step size δ, outputs one logit $C_\phi(s, \delta)_u$ for each signed direction $u \in \mathcal{U}$. The model is trained from the controller's own commands. Each executed option stores its commanded direction $u_t^{\text{cmd}} \in \mathcal{U}$ and step size δ_t. Let $L_t = \min\{H,\ \tau_t^{\text{end}} - t\}$ be the realized duration of the command issued at time t, where H is the waypoint timeout and τ_t^{end} is the first step at which the waypoint is reached or the episode terminates. At this option boundary, we compute the fraction of the commanded displacement that was actually realized:

$$y_t = \text{clip}\left(\frac{\langle (z_{t+L_t} - z_t) \oslash \sigma_g,\ u_t^{\text{cmd}} \rangle}{\delta_t},\ 0,\ 1 \right), \tag{9}$$

where $\oslash$ denotes elementwise division. Like the critic targets in Eq. 15, each label thus reflects only the out-

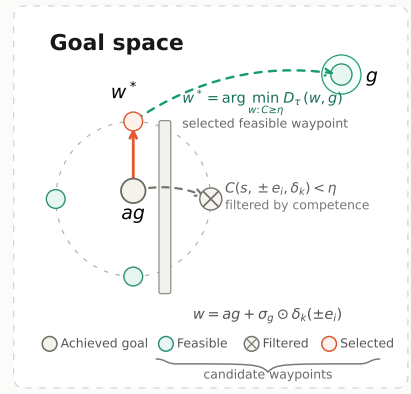


Figure 4: CALP goal-space planning. Candidate subgoals are generated along signed coordinate directions around the achieved goal ag. Candidates the controller is predicted unable to realize ($C < \eta$, shorthand for Eq. 11) are filtered out, and the remaining candidate with the smallest temporal distance to g is selected.

come of a single command. This hindsight target is graded rather than binary: it equals 1 when the commanded displacement is fully realized, 0 when no progress is made along u_t^{cmd}, and takes intermediate values for partial progress. We fit C_ϕ with a sigmoid binary cross-entropy loss on this soft label, applied only to the logit of the commanded direction:

$$\mathcal{L}_C(\phi) = -\,\mathbb{E}_{\mathcal{D}}\Big[y_t \log \sigma(\ell_t) + (1-y_t)\log\big(1-\sigma(\ell_t)\big)\Big], \qquad \ell_t = C_\phi(s_t, \delta_t)_{u_t^{\mathrm{cmd}}}, \tag{10}$$

where $\sigma(\cdot)$ is the logistic function. At planning time, $\sigma\big(C_\phi(s,\delta_k)_u\big)$ predicts the fraction of the displacement toward candidate $z + \sigma_g \odot (\delta_k u)$ that the controller can realize, and we retain only candidates whose predicted competence exceeds a threshold η:

$$\mathcal{W}_C(s) = \big\{\, z + \sigma_g \odot (\delta_k u) \in \mathcal{W}(s) \;:\; \sigma\big(C_\phi(s,\delta_k)_u\big) \geq \eta \,\big\}. \tag{11}$$

In Figure 4, for instance, the candidate behind the wall is removed because the controller cannot realize it.

Learning lower-quantile temporal distance. To rank feasible candidates, we learn a temporal-distance model $D_\psi(z, g)$ that estimates the number of steps needed to reach goal g from achieved goal z. In Section 5.3, we show that replacing it with a full-state model $D_\psi(s, g)$ does not significantly degrade performance, indicating that does D not depend on a low-dimensional input. Training pairs are constructed by hindsight relabeling (Andrychowicz et al., 2017): for a segment from s_t to s_{t+j}, we treat z_{t+j} as the goal and use $d_{t,j} = \log(1+j)$ as the target. Since exploratory trajectories often wander, a squared loss would regress toward these slow paths. We therefore fit a lower conditional quantile via quantile regression (Koenker & Bassett, 1978), using the pinball loss ρ_τ with $\tau_D < 1/2$, which approximates the traversal time of the faster observed paths between z_t and z_{t+j} (Figure 5). This estimate is optimistic by design, and feasibility is handled separately by the competence filter in Eq. 11.

$$\mathcal{L}_D(\psi) = \mathbb{E}_{\mathcal{D}}\Big[\rho_{\tau_D}\big(d_{t,j} - D_\psi(z_t, z_{t+j})\big)\Big], \tag{12}$$

with $\tau_D = 0.2$ in all experiments. The planner then selects the feasible candidate closest to the final goal g:

$$w^\star = \arg\min_{w \in \mathcal{W}_C(s)} D_\psi(w, g). \tag{13}$$

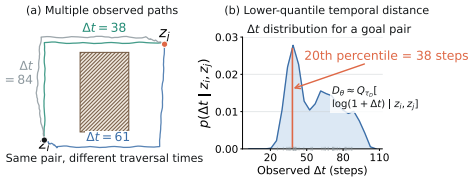


Figure 5: Lower-quantile temporal distance. (a) The same goal pair is connected by paths with different traversal times. (b) Mean regression is pulled toward slow, wandering paths, whereas the 20th percentile tracks the faster observed routes.

Progress-aware local controller. A single controller $\pi_\omega(x_t)$ is shared across all directions and step sizes, with input $x_t = (s_t, w_t, u_t, \alpha_t)$, where w_t is the active waypoint, $u_t \in \mathcal{U}$ the commanded direction, and α_t the steps since w_t was issued. Each waypoint remains active until it is reached or times out after H steps, after which CALP replans. The controller is rewarded for normalized progress along the commanded displacement from the achieved goal z_{start} at which w_t was issued:

$$r_t^w = \big\langle (z_{t+1} - z_t) \oslash \sigma_g,\ \hat{v}_w \big\rangle, \qquad \hat{v}_w = \frac{(w_t - z_{\mathrm{start}}) \oslash \sigma_g}{\|(w_t - z_{\mathrm{start}}) \oslash \sigma_g\|_2}. \tag{14}$$

When the final goal is commanded directly ($w_t = g$), we use the decrease in normalized distance to g instead. Both rewards are computed from achieved goals alone.

To avoid propagating value across different commands, the critics use a truncated multi-step target that never crosses a waypoint boundary. With $h_t \leq H_{\mathrm{TD}}$ the number of consecutive steps from t under the same waypoint,

$$q_t = \sum_{i=0}^{h_t-1} \gamma^i r_{t+i}^w + \mathbf{1}[h_t = H_{\mathrm{TD}}]\,\gamma^{H_{\mathrm{TD}}} \min_{m\in\{1,2\}} Q_{\bar{\theta}_m}\big(x_{t+H_{\mathrm{TD}}},\, \pi_{\bar{\omega}}(x_{t+H_{\mathrm{TD}}})\big), \tag{15}$$

so the target bootstraps only if the waypoint stays active for the full H_{TD} steps and is otherwise truncated at the waypoint's end or episode termination. Each target therefore reflects the outcome of a single command, which keeps credit assignment consistent even as the planner switches waypoints. The controller is updated with the TD3 actor objective (Fujimoto et al., 2018). We use $H = 120$ and $H_{\mathrm{TD}} = 20$ (Appendix B); since $H_{\mathrm{TD}} < H$, most commands last long enough for bootstrapping, while the short TD horizon limits error propagation within each command. Because every waypoint lies near the current achieved goal, the controller only needs to master local goal reaching, and long-horizon behavior emerges from the sequence of planning decisions.

5 EXPERIMENTS

We first compare CALP's goal-reaching performance and training cost against shallow and depth-scaled baselines. Second, we isolate the contributions of the competence filter and the lower-quantile distance objective, and examine how the distance model generalizes from goal-space to full-state inputs. Finally, we vary the network depth to assess whether CALP's performance stems from its planning structure or from model capacity. A sensitivity analysis of CALP's key hyperparameters, the competence threshold η and the TD horizon H_{TD}, is provided in Appendix A.

5.1 SETUP



Figure 6: Visualization of JaxGCRL tasks used in our experiments.

Environments. All RL experiments use JaxGCRL (Bortkiewicz et al., 2025), a suite of GPU-accelerated goal-conditioned continuous-control tasks built on the Brax (Freeman et al., 2021) and MJX (Todorov et al., 2012) simulators. We evaluate on eight locomotion, navigation, and manipulation tasks (Figure 6), all following the standard sparse-reward setting of JaxGCRL. Following Wang et al. (2025), we measure performance as the number of time steps the agent spends near the goal (*time at goal*). When reporting performance as a single number, we average over the last five training epochs, and we report wall-clock time in the same way. All timings are measured on a single NVIDIA RTX 4090 GPU, and all results are averaged over 5 seeds.

Baselines. We compare CALP against six baselines: the contrastive method CRL (Eysenbach et al., 2022) and its depth-scaled variant Scaling CRL (Wang et al., 2025), evaluated at 16 and 64 layers; the online subgoal-based method DCP and SSGC (S K et al., 2026); and TD-based methods TD3 (Fujimoto et al., 2018)combined with hindsight experience replay (Andrychowicz et al., 2017). For all baselines, we use the official or JaxGCRL implementations with the hyperparameters recommended by their authors; details are given in Appendix B.

5.2 PERFORMANCE–COMPUTE TRADE-OFF

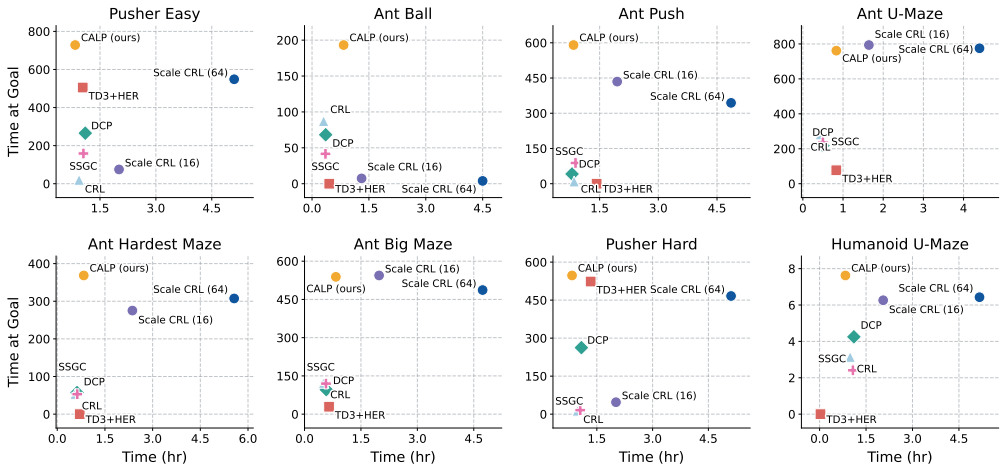


Figure 7: Performance vs. wall-clock training time across eight JaxGCRL domains. Each marker is one method; methods closer to the upper-left corner achieve a better performance–cost trade-off. CALP lies on or near the upper-left frontier in most domains and outperforms all baselines while being several times cheaper than Scaled CRL.

Figure 7 reports results on all eight domains (Bortkiewicz et al., 2025), with the corresponding learning curves in Appendix A. The x-axis shows wall-clock training time on a single RTX 4090 GPU, and the y-axis shows performance, measured as time steps spent at the goal. Methods toward the upper left (↖) are more compute-efficient. CALP achieves the highest performance in six of the eight domains and matches the deep Scaling CRL variants in the remaining two (Ant U-Maze and Ant Big Maze), while requiring a fraction of their training time. Among methods with comparable training cost, CALP outperforms all baselines in every domain, making it the most compute-efficient method overall.

5.3 Ablations

We ablate five design choices: (i) removing the competence estimator C_ϕ (w/o C); (ii) removing the temporal-distance model D_ψ, selecting among feasible candidates either uniformly at random (Uniform) or by Euclidean distance to the goal, $\arg\min_w \|w-g\|_2$ (arg min); (iii) replacing the goal-space distance model $D_\psi(z, g)$ with a full-state model $D_\psi(s, g)$ ($z \to s$); (iv) replacing the quantile loss (Koenker & Bassett, 1978) for temporal distance with a mean-squared-error loss (Quantile → MSE); and (v) varying the depth of the networks used by CALP. Table 1 summarizes the results. The competence filter and the distance model are both essential in obstacle-rich tasks, the quantile objective improves performance in every domain, the distance model remains effective with full-state inputs, and CALP does not require deep networks to perform well.

Table 1: Ablation study and network depth analysis. Time at goal (↑), averaged over 5 seeds. Bold indicates the best result within each group.

	Component ablation						**Network depth**			
Env	CALP	w/o C	w/o D_ψ (Uniform)	w/o D_ψ (arg min)	$z \to s$	Quantile → MSE	2	4	8	16
Ant Ball	193.2	**196.0**	182.5	191.0	188.4	144.5	193.2	211.2	**226.4**	204.3
Ant Hardest Maze	**368.5**	298.1	99.5	173.2	341.2	281.8	368.5	**414.4**	398.6	382.7
Pusher Hard	**547.3**	543.3	525.0	533.5	511.6	454.3	547.3	533.2	539.6	**552.3**
Ant Push	**590.5**	365.9	189.0	224.4	572.7	329.1	—	—	—	—

Competence and quantile distance both matter. Figure 8 shows the competence and quantile ablations on four domains. The effect of removing the competence filter depends on environment geometry. In Pusher Hard and Ant Ball, where local motion is rarely obstructed, performance is essentially unchanged. In Ant Hardest Maze and Ant Push, which require awareness of obstacles, performance drops. Because the lower-quantile distance is deliberately optimistic, a candidate behind a wall can appear closest to the goal. Without the filter, the planner selects it, the controller pursues it until the command times out, and with the achieved goal nearly unchanged, the planner selects the same infeasible waypoint again, trapping the agent in repeated failed commands. Replacing the quantile loss with MSE degrades performance in every domain. MSE pulls distance estimates toward the long traversal times of wandering, relabeled trajectories, whereas the lower quantile tracks the faster observed routes, giving a sharper signal for ranking candidates.

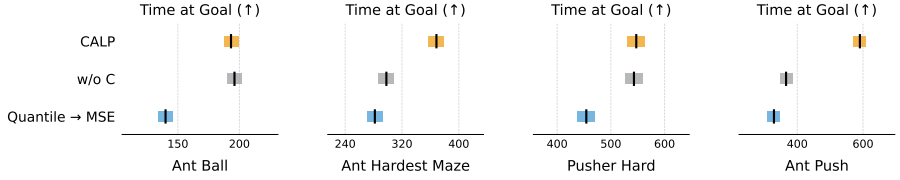


Figure 8: Component ablations on four domains. Removing the competence estimator (w/o C) sharply degrades performance in geometry-sensitive tasks (Ant Hardest Maze, Ant Push) while leaving simpler tasks largely unaffected. Replacing the quantile loss with MSE degrades performance in every domain. Bars show the mean, and shaded regions show the standard deviation over 5 seeds.

The learned distance captures connectivity. To isolate the role of D_ψ, we keep the candidate set and competence filter fixed and replace only the selection rule, choosing among feasible candidates either uniformly at random or by Euclidean distance to the goal. In obstacle-rich tasks, the learned distance is essential. On Ant Hardest Maze and Ant Push, uniform selection reduces time at goal by more than half, and Euclidean selection recovers only part of this gap, leaving CALP more than $2\times$ better (Figure 9). Euclidean selection fails for the reason identified in Section 4.1. In open tasks, the three rules perform nearly identically, since CALP commands the goal directly once it is close. The learned distance thus matters most where geometric proximity and temporal reachability diverge.

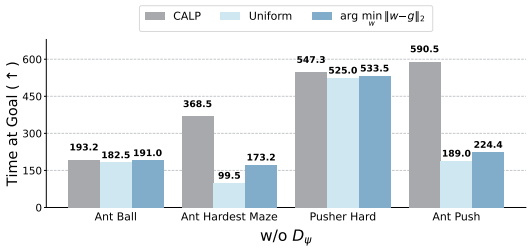


Figure 9: Selection rules among feasible candidates: random (Uniform) or by Euclidean distance to the goal, $\arg\min_w \|w - g\|_2$ (arg min). The learned distance matters most in obstacle-rich tasks.

CALP remains effective with full-state distance inputs. Figure 10 replaces the goal-space distance model $D_\psi(z, g)$ with a full-state model $D_\psi(s, g)$ in Ant Hardest Maze and Pusher Hard. Although $D_\psi(s, g)$ must extract goal-relevant structure from a much higher-dimensional input, performance changes only marginally. This suggests that, once the controller reliably reaches nearby goals, the resulting coverage is sufficient for a shallow network to learn temporal distances from the full state as well, and that CALP's effectiveness stems from its planning structure rather than from the low dimensionality of the goal space.

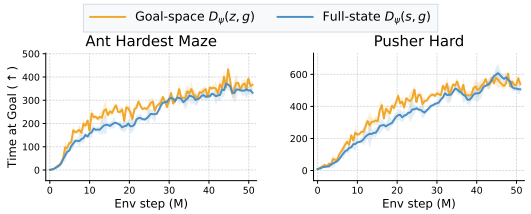


Figure 10: Goal-space (z) versus full-state (s) inputs for the temporal-distance model. Performance is nearly unchanged in both domains.

CALP does not require deep networks for strong performance. Figure 11 varies the depth of the networks used by CALP. Gains from additional depth are modest and inconsistent: the best depth differs across domains (8 for Ant Ball, 4 for Ant Hardest Maze, and 16 for Pusher Hard), the largest improvement over depth 2 is Ant Ball, and Pusher Hard remains within 3% of the default at every depth. Even the default depth-2 model is competitive with the 16- and 64-layer Scaling CRL variants, at a fraction of the training cost. This suggests that CALP's performance stems mainly from its planning structure rather than from network capacity.

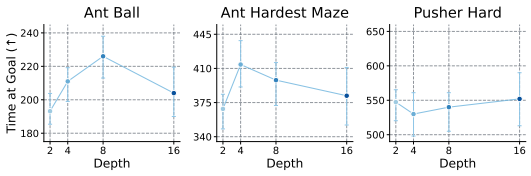


Figure 11: Effect of network depth on CALP. Gains from depth are modest and inconsistent across domains.

6 Conclusion and Limitations

This work asked whether shallow networks, given an appropriate planning structure, can match the long-horizon goal-reaching performance that recent work attains through depth. CALP answers this affirmatively by reaching distant goals through repeated, reliably executed local decisions: with two-hidden-layer networks, it outperforms 16- and 64-layer contrastive agents on six of eight JaxGCRL tasks and matches them on the rest, while training about five times faster. We note two limitations. First, its absolute performance on Humanoid U-Maze remains low, likely because the goal space omits the balance and posture that determine reachability, and because stable locomotion must be learned before any waypoint can be reached. Second, axis-aligned candidates assume a low-dimensional, locally Euclidean goal space and do not extend directly to high-dimensional goals such as images. However, natural images are thought to lie near low-dimensional manifolds (Pope et al., 2021), suggesting that CALP's local planning could carry over by proposing candidates and measuring distances in a learned low-dimensional latent space, which we leave for future work.

AI USE STATEMENT

In this work, we used generative AI tools for debugging code. We have not used generative AI tools for generating the core research ideas, formulating the proposed method; all of these were conceived and developed entirely by the authors. Additionally, we used generative AI tools for improving the layout and visual presentation of figures. We have reviewed all AI-assisted work. Specifically, all code fixes suggested by AI tools were manually inspected, and the affected experiments were re-run and verified by the authors; and AI-assisted figure adjustments were limited to formatting and presentation, with all plotted data and results verified against our original experimental outputs. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not raise any concerns with respect to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

Experimental settings, including hyperparameters, model architectures, and computing resources, are provided in Appendix B.

REFERENCES

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Onur Beker, Mohammad Mohammadi, and Amir Zamir. Palmer: Perception-action loop with memory for long-horizon planning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Diana Borsa, André Barreto, John Quan, Daniel Mankowitz, Rémi Munos, Hado van Hasselt, David Silver, and Tom Schaul. Universal successor features approximators. In *International Conference on Learning Representations (ICLR)*, 2019.

Michał Bortkiewicz, Władysław Pałucki, Vivek Myers, Tadeusz Dziarmaga, Tomasz Arczewski, Łukasz Kuciński, and Benjamin Eysenbach. Accelerating goal-conditioned reinforcement learning algorithms and research. In *International Conference on Learning Representations (ICLR)*, 2025. Spotlight.

Konrad Czechowski, Tomasz Odrzygóźdź, Marek Zbysiński, Michał Zawalski, Krzysztof Olejnik, Yuhuai Wu, Łukasz Kuciński, and Piotr Miłoś. Subgoal search for complex reasoning tasks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 624–638, 2021.

Will Dabney, Mark Rowland, Marc G. Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. Search on the replay buffer: Bridging planning and reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. C-learning: Learning to achieve goals via recursive classification. In *International Conference on Learning Representations (ICLR)*, 2021.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax – a differentiable physics engine for large scale rigid body simulation. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning (ICML)*, 2018.

Danijar Hafner, Kuang-Huei Lee, Ian Fischer, and Pieter Abbeel. Deep hierarchical planning from pixels. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Kristian Hartikainen, Xinyang Geng, Tuomas Haarnoja, and Sergey Levine. Dynamical distance learning for semi-supervised and unsupervised skill discovery. In *International Conference on Learning Representations*, 2020.

Christopher Hoang, Sungryull Sohn, Jongwook Choi, Wilka Carvalho, and Honglak Lee. Successor feature landmarks for long-horizon goal-conditioned reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Edward S. Hu, Richard Chang, Oleh Rybkin, and Dinesh Jayaraman. Planning goals for exploration. In *International Conference on Learning Representations*, 2023.

Leslie Pack Kaelbling. Learning to achieve goals. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1094–1098, 1993.

Roger Koenker and Gilbert Bassett, Jr. Regression quantiles. *Econometrica*, 46(1):33–50, 1978. doi: 10.2307/1913643.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.

Hojoon Lee, Dongyoon Hwang, Donghu Kim, Hyunseung Kim, Jun Jet Tai, Kaushik Subramanian, Peter R. Wurman, Jaegul Choo, Peter Stone, and Takuma Seno. Simba: Simplicity bias for scaling up parameters in deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2025.

Andrew Levy, George Konidaris, Robert Platt, and Kate Saenko. Learning multi-level hierarchies with hindsight. In *International Conference on Learning Representations (ICLR)*, 2019.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Bo Liu, Yihao Feng, Qiang Liu, and Peter Stone. Metric residual networks for sample efficient goal-conditioned reinforcement learning. In *AAAI Conference on Artificial Intelligence*, 2023.

Minghuan Liu, Menghui Zhu, and Weinan Zhang. Goal-conditioned reinforcement learning: Problems and solutions. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI), Survey Track*, pp. 5502–5511, 2022.

Vivek Myers, Chongyi Zheng, Anca Dragan, Sergey Levine, and Benjamin Eysenbach. Learning temporal distances: Contrastive successor features can provide a metric structure for decision-making. In *International Conference on Machine Learning (ICML)*, 2024.

Ofir Nachum, Shixiang Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

Soroush Nasiriany, Vitchyr H. Pong, Steven Lin, and Sergey Levine. Planning with goal-conditioned policies. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Michal Nauman, Mateusz Ostaszewski, Krzysztof Jankowski, Piotr Miłoś, and Marek Cygan. Bigger, regularized, optimistic: Scaling for compute and sample-efficient continuous control. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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 (NeurIPS)*, 2023.

Seohong Park, Kevin Frans, Deepinder Mann, Benjamin Eysenbach, Aviral Kumar, and Sergey Levine. Horizon reduction makes rl scalable. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Silviu Pitis, Harris Chan, Stephen Zhao, Bradly Stadie, and Jimmy Ba. Maximum entropy gain exploration for long horizon multi-goal reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7750–7761. PMLR, 2020.

Phillip Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. The intrinsic dimension of images and its impact on learning, 2021. URL `https://arxiv.org/abs/2104.08894`.

Zhifeng Qian, Mingyu You, Hongjun Zhou, Xuanhui Xu, and Bin He. Goal-conditioned reinforcement learning with disentanglement-based reachability planning. *IEEE Robotics and Automation Letters*, 2023.

Swaminathan S K, Damiya Gondha, Theyanesh Eswaramoorthy Rajahkrishnan, and Aritra Hazra. Direction-conditioned policies via compositional subgoal scoring for online goal-conditioned reinforcement learning, 2026. 2nd Workshop on Compositional Learning at ICML 2026.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1312–1320. PMLR, 2015.

Max Schwarzer, Johan Obando-Ceron, Aaron Courville, Marc G. Bellemare, Rishabh Agarwal, and Pablo Samuel Castro. Bigger, better, faster: Human-level atari with human-level efficiency. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pp. 30365–30380. PMLR, 2023.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International Conference on Machine Learning*, pp. 387–395, 2014.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2017.

Kevin Wang, Ishaan Javali, Michał Bortkiewicz, Tomasz Trzciński, and Benjamin Eysenbach. 1000 layer networks for self-supervised rl: Scaling depth can enable new goal-reaching capabilities. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. Best Paper Award.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 36411–36430. PMLR, 2023.

Tianren Zhang, Shangqi Guo, Tian Tan, Xiaolin Hu, and Feng Chen. Generating adjacency-constrained subgoals in hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21579–21590, 2020.

Chongyi Zheng, Benjamin Eysenbach, Homer Walke, Patrick Yin, Kuan Fang, Ruslan Salakhutdinov, and Sergey Levine. Stabilizing contrastive rl: Techniques for robotic goal reaching from offline data. In *International Conference on Learning Representations (ICLR)*, 2024a.

Chongyi Zheng, Ruslan Salakhutdinov, and Benjamin Eysenbach. Contrastive difference predictive coding. In *International Conference on Learning Representations (ICLR)*, 2024b.

A Additional Experimental Results

A.1 Learning Curves

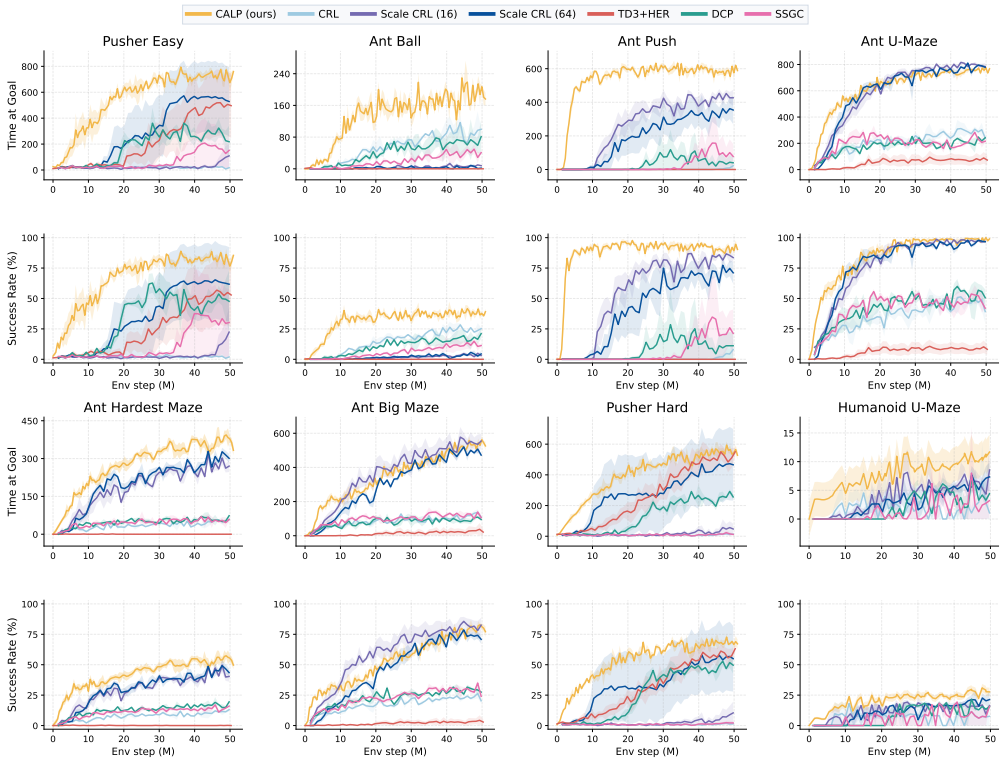


Figure 12: Learning curves across all eight environments. We compare CALP with CRL, scaled CRL variants of depths 16 and 64, and the goal-conditioned RL baselines TD3+HER, DCP, and SSGC. For each environment, we report Time at Goal (top) and success rate (bottom) as functions of environment interactions. CALP achieves strong performance and learning efficiency across environments without relying on substantially deeper networks. Higher is better for both metrics.

A.2 Sensitivity to the competence threshold and competence-label horizon.

Figure 13 studies the two parameters that most directly control the planner's behavioral trade-off. We first vary the competence threshold $\eta \in \{0.3, 0.5, 0.7\}$. Lower values admit more ambitious but less reliable waypoints, whereas higher values are more conservative. We then vary the competence-label horizon $H_{\mathrm{TD}} \in \{10, 20, 40\}$. A short horizon produces local but potentially noisy progress targets, whereas a long horizon captures more of the option outcome but is more frequently truncated at option boundaries. Importantly, H_{TD} only defines the learning target and does not determine when the controller replans; option execution is capped separately by $T_{\mathrm{opt}} = 120$. We use $\eta = 0.5$ and $H_{\mathrm{TD}} = 20$ across all environments without environment-specific tuning.

B Reproducibility

Hardware. All training ran on a machine with an Intel(R) Core(TM) i9-13900K CPU with 24 cores (8 performance + 16 efficiency cores), base clock at 3.0 GHz (boost up to 5.8 GHz) with 32 GB RAM and an NVIDIA GeForce RTX 4090 24 GB GPU. Ablation experiments on depth ran on an A100 on cluter.

Hyperparameters and training scripts. All shared hyperparameters follow the defaults of JaxGCRL (Bortkiewicz et al., 2025) and DCP (S K et al., 2026): discount factor $\gamma = 0.99$, un-

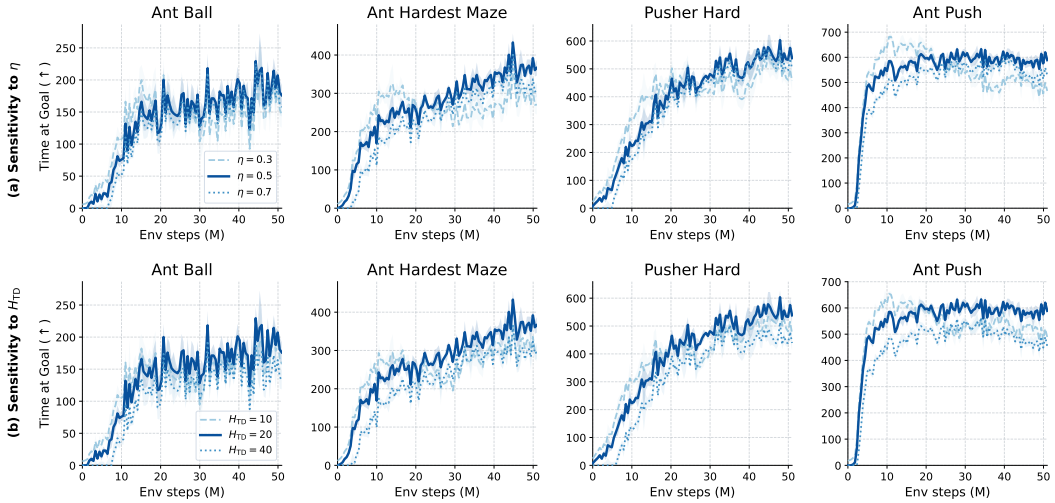


Figure 13: Sensitivity to the competence threshold and competence-label horizon. Top: time at goal for $\eta \in \{0.3, 0.5, 0.7\}$. Bottom: time at goal for $H_{\mathrm{TD}} \in \{10, 20, 40\}$. Shading denotes standard error over five seeds. The default setting ($\eta = 0.5, H_{\mathrm{TD}} = 20$) is shared across all environments.

roll length 62, 512 parallel environments, batch size 256, and episode length 1000. For CALP, unless stated otherwise, we use competence-label horizon $H_{\mathrm{TD}} = 20$, option timeout $T_{\mathrm{opt}} = 120$, two normalized displacement scales $\Delta = \{0.5, 1.0\}$, competence threshold $\eta = 0.5$, distance quantile $\tau_D = 0.2$, and direct-goal threshold $\epsilon_{\mathrm{dir}} = 0.5$. Both the competence and temporal-distance models use learning rate 3×10^{-4}. The replay buffer contains at most 50,000 transitions, and parameter updates begin after 1,000 transitions. A selected waypoint is executed until it is reached or the option timeout is reached; if the waypoint is reached early, CALP immediately replans from the current state. We report means over five seeds. Code is available at `https://anonymous.4open.science/r/calp-anonymous-source-818F/`.