LATENT SKILL ACQUISITION AND COMPOSITION FOR CONTACT-RICH MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Human-level generalization in manipulation requires composing known behaviors to reach novel goals in scenarios never seen in training. The challenge sharpens in contact-rich settings, where behaviors must be recomposed at the granularity of contact rather than of a semantic subtask. Without relying on per-task demonstrations, a promising approach is to leverage test-time compute to perform *physical reasoning* that searches for novel solutions and improves them incrementally during deployment. To effectively perform search in the large space of motions, we propose SkillReasoner, which learns temporal abstractions of actions (i.e., skills) from unstructured offline interactions, and then reasons over and recomposes them at test time. Crucial to the formulation is a latent skill space learned with varying temporal horizons, so that composition can happen at varying granularity, as required by different stages of the task, while minimizing the number of decisions. To realize a novel goal at test time, the robot searches over latent skill sequences in its imagination, decodes a chosen skill into low-level actions, and replans in closed loop, up to 30 Hz. We demonstrate on a series of contact-rich real-world and simulated tasks that the robot chains behaviors to achieve novel goals that require far longer behaviors than those seen in training, and adapts zero-shot to unseen perturbations, changing dynamics, and constraints, all trained purely from unstructured play data without task-specific demonstrations. Project website: `https://skillreasoner-anon.github.io/`.

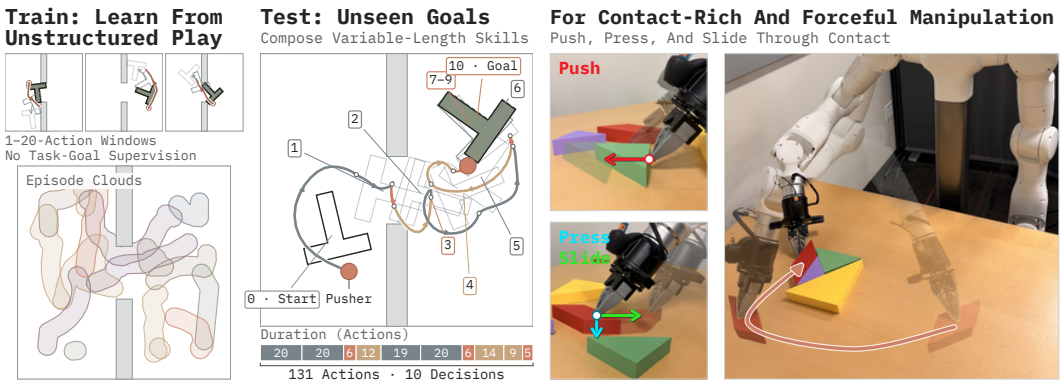

Figure 1: **Acquiring and composing latent skills of varying durations enables generalization to unseen goals in contact-rich tasks.** SkillReasoner acquires a latent skill space with varying temporal horizons from unstructured play data, then re-composes them at varying granularities with test-time search, using long motions for progress and short motions for fine adjustments. By building its "behavior vocabulary" from physical interactions, these skills capture reusable low-level patterns that can be composed to achieve unseen goals in real-world contact-rich and forceful tasks.

1 INTRODUCTION

Achieving open-world robots with human-level behaviors demands handling the infinite long tail of the real world, generalizing across every possible (state $\times$ goal) an agent may encounter. While behavior cloning has led to substantial progress in acquiring dexterous behaviors from expert demonstrations (Zhao et al., 2023; Chi et al., 2023), it remains unclear how to achieve human-level gener-

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.

alization in contact-rich settings, especially those involving novel goals and unseen compositions of motions beyond what has been demonstrated.

Rather than scaling learning to represent *given* solutions, a key alternative is to use test-time compute to search for *novel* solutions and distill them iteratively at deployment, as evidenced in applications such as AlphaGo (Silver et al., 2016) and reasoning-based agentic coding systems. Yet manipulation suffers from a combinatorially large search space, which makes search challenging, if not intractable, when carried out with low-level actions. Existing approaches tackle this by indexing skills with language (Huang et al., 2022; Ahn et al., 2022) or pre-defining a skill repertoire (Kaelbling & Lozano-Pérez, 2011), and then delegate skill compositions to a language model or a discrete planner to compose behaviors for a novel goal. Yet the expressive power of composition is determined by its lowest-level building blocks. For contact-rich manipulation, many useful building blocks lie below the level of semantic behaviors that humans name or specify, making semantic or linguistic skill spaces too coarse for composing novel interactions.

This problem calls for the need for *latent skill acquisition and composition*: learn to discover a "behavior vocabulary" (i.e., temporal abstractions of skills) in an *unsupervised* way from *unstructured* data, and then reason over and re-compose them at test time given a novel goal. Critically, the abstractions should capture nuanced low-level patterns, including those involving contact and force, from data gathered by interacting with the world without prescribing in advance what those patterns should be. However, a key challenge in skill learning is trading off horizon reduction in decision making for the freedom to recombine learned behaviors: longer skills reuse coordinated sequences from the data as single decisions, while shorter skills expose finer interactions for new compositions but suffer more from compounding errors. Since skill horizon determines the temporal extent over which behavior is reused as a unit, we propose to learn a latent skill space with *varying* horizons, so that the robot can reuse learned behaviors at the granularity required by a given task (e.g., repurpose shorter behaviors for fine adjustments near clearance and final alignment in Fig. 1).

In **SkillReasoner**, we formulate skill learning as a representation learning problem: given unstructured interaction data, we learn a latent skill space that reconstructs a varying temporal horizon of actions conditioned on observations, accompanied by (i) a skill decoder that decodes a latent skill into a sequence of actions, (ii) a skill dynamics model that forward-simulates the environment in latent state space under a given latent skill, and (iii) a skill prior that samples varying-horizon skills given observations. At test time, given a novel goal specification, the robot searches many steps into the future purely in imagination by sampling candidate skills from the skill prior and simulating their effects with the skill dynamics, decodes the most promising immediate skill into low-level actions with the skill decoder, executes them in the environment, and repeats with closed-loop replanning.

Viewed as a hierarchical policy, SkillReasoner separates *task-specific* composition by a search-based high-level policy from *task-agnostic* dexterous behavior execution by a learned low-level policy. This enables achiving new goals and satisfying unseen constraints through reasoning with the vocabulary of a learned "behavior language", instead of natural language that can be cumbersome to describe motions and contact. We further design an efficient architecture for the skill dynamics and the skill prior, which supports closed-loop replanning up to 30 Hz with a consumer GPU.

We summarize our contributions as follows: (i) we formulate generalization in contact-rich manipulation as a latent skill composition problem with varying-horizon skills and propose a formulation that learns from unstructured offline interactions; (ii) we design an efficient system implementation that supports closed-loop replanning and zero-shot generalization to novel goals unseen from training for a range of challenging simulated and real-world contact-rich tasks; and (iii) we demonstrate various appealing properties of the system including adaptation to test-time constraints, perturbation recovery, changing dynamics, and self-improvement with few-shot interactions.

2 Related Work

Hierarchical RL addresses decision-making problems through reusable skills and a mechanism for selecting them (Kaelbling, 1993; Sutton et al., 1999; Precup, 2000; Dietterich, 2000). Some skill-discovery methods find subgoals at states common to successful trajectories or at bottlenecks in a graph of observed transitions (McGovern & Barto, 2001; Menache et al., 2002), while others chain options backward from a target (Konidaris & Barto, 2009). Other approaches learn skills jointly with

a reward-driven high-level policy (Bacon et al., 2017; Harb et al., 2018), use intrinsic objectives that favor behavioral diversity (Eysenbach et al., 2019) or predictable outcomes (Sharma et al., 2020), or extract them from demonstrations (Konidaris et al., 2012; Zhu et al., 2022). Similar to prior works that learn skills by compressing offline experience (Ajay et al., 2021; Jiang et al., 2022), we learn skills by reconstructing actions from play data, with the low-level policy trained on human-recorded trajectories that has enabled recent advancements in learning dexterous behaviors (Gemini Robotics Team et al., 2025; Black et al., 2025; TRI LBM Team et al., 2026). Formulating skill acquisition as compression from offline data avoids solving downstream credit assignment in skill learning but entails requiring sufficient coverage of latent state space and local behaviors. Apart from low-level skills, skill selectors can be either a planner that chooses skills by evaluating their effects (Xie et al., 2021; Nie et al., 2026) or a high-level policy that maps states and goals to skills. Konidaris et al. (2018) derive planning symbols from the initiation and effects of given options, while SymSkill learns object-relative predicates, operators, and motion skills from unsegmented demonstrations (Shao et al., 2026). TACO-RL and HIQL instead train a high-level policy through offline RL with hindsight-relabeled goals (Rosete-Beas et al., 2023; Park et al., 2023). We use a search-based planner for skills as it enables adaptive computation to achieve goals and satisfy constraints not seen during training; the learned skill space could also support subsequent online learning (Pertsch et al., 2021). Closely related, SkiMo (Shi et al., 2023) learns a latent skill space and a skill dynamics model to accelerate learning in online RL. In this work, we focus on contact-rich manipulation, which demands recomposition of both long coordinated motions and fine adjustments, and propose learning skills with varying horizons to address this challenge.

3 METHOD

We motivate the goal of building a composable varying-horizon latent skill space in Section 3.1, discuss how to do inference with it in Section 3.2, and describe our instantiation in Section 3.3.

3.1 SKILLREASONER: VARYING-HORIZON LATENT SKILL COMPOSITION

We seek latent skills that can be recomposed at the temporal granularity required by a new task, without prescribing their boundaries or semantics. From play data containing no task labels or rewards, we sample trajectory windows of varying duration d and encode each as a skill u. Long windows capture extended interactions as single decisions, while short windows expose intermediate states for fine-grained continuations. Varying the horizon therefore supports efficient progress and fine-grained composition in the same skill space.

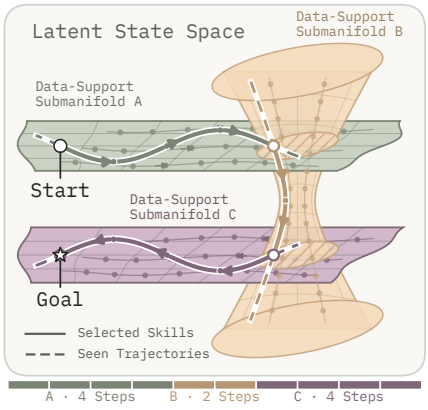

Figure 2: **The Idea.** Play data supplies local interactions and observed transitions and covers submanifolds of latent state space. By exposing varying-horizon segments of these interactions as latent skills, SkillReasoner composes them to reach unseen goals, where the full trajectory may not be in the training data.

Figure 2 provides a conceptual illustration of the resulting skill space as an implicit, data-supported roadmap (Kavraki et al., 1996; Hauser & Latombe, 2008). Intuitively, locally meaningful interactions in the data cover submanifolds of the latent state space, expose learned clusters of their varying-length segments as latent skills, and provide opportunities to chain them into a sequence for a test-time task unseen during training. The illustration also shows, informally, that the extent of each submanifold in latent state space depends on how much locally meaningful behavior the play data contain (less structured data may require more chaining for the same trajectory), that chaining requires compatible neighborhoods between adjacent submanifolds, and that observed and planned states should stay in regions supported by the data. This formulation is thus particularly suited to human-collected behaviors, whose temporally correlated interactions tend to concentrate in narrow regions of state–action space, providing coherent local behaviors for skill learning and composition without explicit task structure. Notably, when data contain behaviors from only one source task, this interpretation

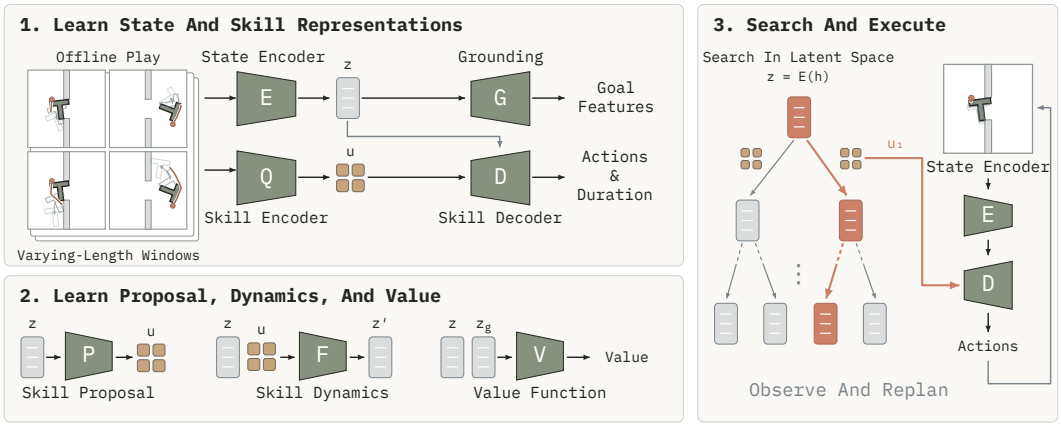

Figure 3: **SkillReasoner Overview**, where z, z', and z_g denote latent states for current, next, and goal observations and u denotes a latent skill. We first learn state and varying-horizon skill representations from offline play, then learn the proposal, dynamics, and value function in the latent state and skill space. At test time, state encoder encodes latest observation–action history h into latent state, beam search plans a sequence of latent skills, skill decoder executes the first selected skill u_1, and then replans using new observations.

also shows that SkillReasoner can train on mixed-quality data rather than only successful demonstrations: failed trajectories can still contribute useful local windows.

3.2 Inference with SkillReasoner

To achieve an unseen goal specified at test time, SkillReasoner acts as a hierarchical policy consisting of (i) a high-level policy that conditions on a goal and outputs next latent skill and (ii) a low-level policy that translates the latent skill into actions.

We use beam search as the high-level policy in the latent skill space. Denoting the current latent state as z_0, we alternate between (i) sampling candidate skills from a proposal function $P : z \mapsto p_P(u \mid z)$, and (ii) rolling them forward with a dynamics function $F : (z, u) \mapsto \hat{z}'$. Each imagined trajectory is scored by a goal evaluation function c_{g} at its endpoint, a risk proxy c_{r} along its rollout, and a learned cost-to-go c_{h} estimating remaining low-level actions. Formally, for a goal specification g and a candidate sequence $u_{1:\ell}$, we roll out $z_i = F(z_{i-1}, u_i)$ and score

$$J_\ell = c_{\mathrm{g}}(z_\ell, g) + \beta \sum_{i=1}^{\ell} c_{\mathrm{r}}(z_i) + \lambda c_{\mathrm{h}}(z_\ell, g), \tag{1}$$

Here β and λ are nonnegative values weighting risk and the learned cost-to-go. For goal evaluation, we use 3D object keypoints (2D in planar tasks) as compact, composable goal features that can specify a set of goals rather than a single point [1]. Since search is performed in skill space, a rollout can span $\sum_i d_i$ low-level action steps with respective skill durations $d_{1:\ell}$, while needing only ℓ search decisions. After obtaining the lowest-cost skill sequence, the low-level policy converts the first skill into actions with predicted duration. Then the robot executes these actions open-loop, observes the environment, and replans in closed loop. By exposing varying-horizon skills to the search process, the planner adaptively selects temporal granularity for each stage of the task, using long skills to make progress in well-supported regions and short skills when finer composition is needed. Search settings are summarized in Table 4.

3.3 Model Instantiation and Design Choices

We train SkillReasoner on offline play trajectories $\mathcal{D} = \{\tau^n\}_{n=1}^N$, where $\tau = (o_0, a_0, o_1, \ldots, o_T)$ and o_t contains RGB images and proprioception (Figure 3).

[1] At deployment, G predicts keypoints from raw perception and imagined latent states to score against the operator-specified goal, without an external tracker; ground-truth keypoints are used only for training supervision. Alternatively, one can use latent state distance as goal evaluation (Nair et al., 2018), but we find keypoints provide more precise evaluation for contact-rich tasks in our instantiation.

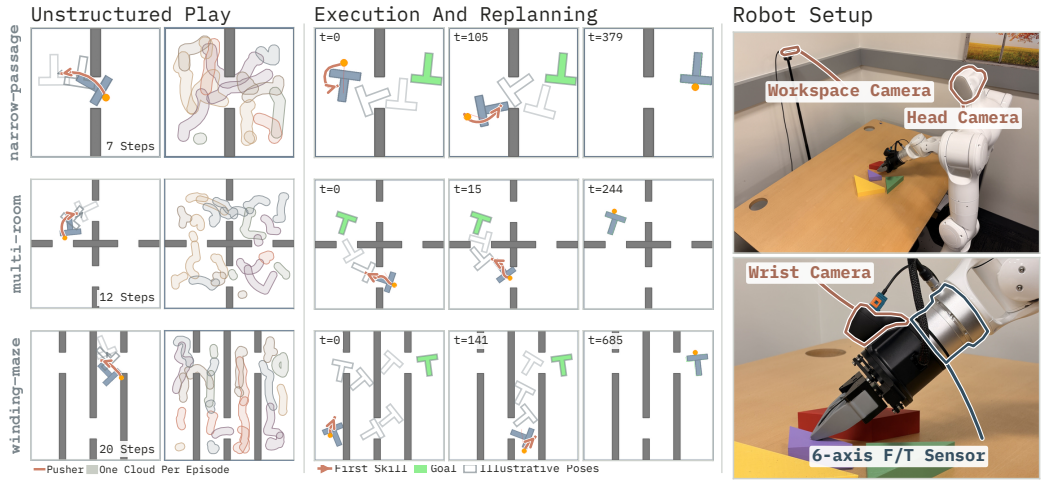

Figure 4: **Evaluation Environments and Robot Setup.** (Left) unstructured play windows and episode-level object trajectory clouds (downsampled for clarity). (Center) successive execution and replanning snapshots show skill composition toward green goal poses. (Right) real-world evaluation setup, where a robot performs contact-rich manipulation with visual perception and a 6-axis force/torque sensor.

State and Skill Representations. We learn the latent state through goal feature prediction and latent skills through an observation-conditioned action reconstruction objective. The state encoder E encodes a short window of 3-step observation–action history h_t to latent state z_t, and the state grounding G predicts goal features $\hat{y}$ from it (keypoints in our work). G also predicts coordinate-wise variance $\nu_y(z_t)$ in squared feature units, which weights feature errors during training and is aggregated into the search risk proxy $c_{\mathrm{r}}(z) = \rho(\nu_y(z))$. To train the skill autoencoder, we sample interaction windows $w_{t:t+d}$ of duration $d \sim \mathrm{Unif}\{1, \ldots, H\}$. The skill encoder Q compresses each window into a fixed-length sequence of discrete tokens u. The skill decoder $D(z_t, u)$ reconstructs its actions and predicts the duration used to terminate the skill. The observations in the window let Q encode what the actions achieved, while conditioning D on z_t adapts the decoded actions to the current scene. Like state grounding G, skill decoder D predicts coordinatewise variance without uncertainty labels; we find that this regularization shapes the learned representations and improves generalization. Contrary to our initial hypothesis, we find random windows outperform hand-defined event boundaries, possibly because variance-weighted training can accommodate their broader set of overlapping continuations. Architecture and loss details are in Appendix C.

Proposal, Dynamics, and Value. We freeze the representation modules E, Q, D, and G, then learn the search components on the resulting latent states and skills. The proposal P learns to sample Q-encoded skills from a state z; the dynamics $F(z, u)$ predicts their encoded endpoints $E(h_{t+d})$; and the value function $V(z, z_g)$ learns negative, duration-aware goal-reaching cost from hindsight goals z_g. We find that training these three components on frozen representations improves performance over joint training and lowers training cost, as they do not ingest raw high-dimensional data. Because F is trained on individual skill transitions but rolled out recursively during search, we use spectral normalization to stabilize its predictions (Miyato et al., 2018). Even though both learned value and object keypoints guide search, we find that they provide complementary roles: value identifies promising regions beyond the search horizon, while direct comparison of keypoint-based goal features often supports precise, fine-grained behaviors that complement the locally inaccurate value estimates. Training objectives and value targets are detailed in Appendix C.

4 Evaluation Setup

Simulation. To study generalization in contact-rich goal reaching, we extend Push-T (Chi et al., 2023) to goal-conditioned, maze-like layouts (Figure 4). A policy uses RGB observations to control a circular pusher at 10 Hz and maneuver a T-shaped object through narrow passages to a target pose. The three environments test complementary capabilities: `narrow-passage` combines extended free-space motion with fine adjustments near the passage and final goal; `multi-room` tests multimodality, with two possible object pathways to the same goal; and `winding-maze` requires

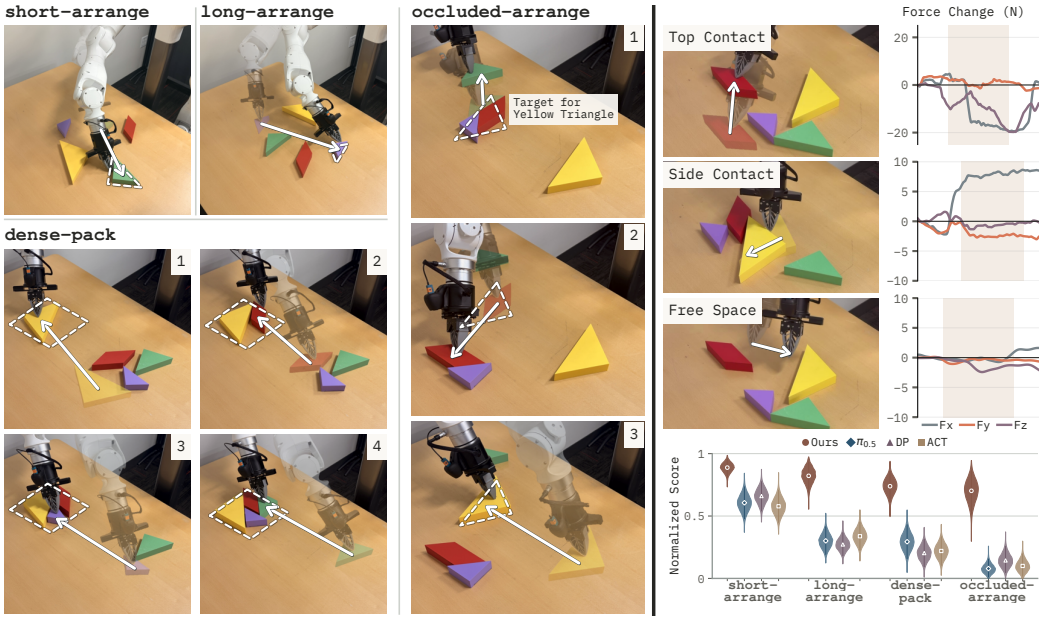

Figure 5: **Real-World Tasks, Learned Skills, and Evaluation.** (Left) four real-world tasks focusing on local alignment, longer-range transport, dense packing, and occluded rearrangement; dashed outlines mark goals. (Upper right) learned skills spanning top-contact sliding, side-contact pushing, and free-space motions, with corresponding force readings for reference. (Bottom right) normalized task scores over ten trials for each entry.

long-horizon rollouts through successive passages. We generate play data by transforming and replaying action sequences from the original Push-T demonstrations in these layouts. Each training episode therefore contains interactions confined to a small region (the episode clouds in Figure 4, left), while test goals require distant transport through one or more passages and final alignment. Training episodes have median lengths of roughly 100 steps. In comparison, typical rollouts to successfully complete the tasks require approximately 250, 300, and 500 steps, respectively for each environment. Beyond long-horizon composition, these tasks also feature discontinuous dynamics, alternating between free-space motion and various types of contact among the pusher, block, and walls, and precise maneuvers around passage clearance and final alignment (requiring 95% object–goal overlap). We invite readers to try these tasks in our anonymous interactive browser demo.

Baselines and Evaluation. We compare against goal-conditioned offline RL with IQL (Kostrikov et al., 2022) and HIQL (Park et al., 2023) using implementations from OGBench (Park et al., 2025a), and behavior cloning with ACT (Zhao et al., 2023), Diffusion Policy (Chi et al., 2023), and $\pi_{0.5}$ (Black et al., 2025).[2] The primary baselines learn from the same play data as SkillReasoner, where the behavior cloning baselines use achieved future poses as hindsight goals. We also provide separate privileged behavior cloning baselines trained on 50 expert demonstrations from the test task distributions. We evaluate on 100 shared reset–goal tasks per entry, ending each trial at 95% object–goal overlap or a maximum of 720 environment steps. Normalized task score assigns 50% to final object–goal overlap and 50% to normalized reduction in remaining nominal-route distance. The route uses L_2 segment lengths through passages to account for walls, rather than direct object–goal distance (used only for evaluation). The one- and two-crossing `winding-maze` entries report maximum passage-clearance progress; the full entry reports the terminal task score.

Real-World Setup. To study SkillReasoner in a controlled and reproducible real-world setup, we control a Tianji Marvin robot at 10 Hz to manipulate 3D-printed objects of varying shapes. The robot is equipped with a wrist-mounted force/torque sensor and onboard and external cameras for visual perception. We deliberately construct tasks that require chaining coordinated contact-rich and forceful behaviors, including top-contact sliding, side-contact pushing, and repeated interactions among

[2]We also experimented with Fable 5.1 and GPT-6-Astra and found promising behavior in our 2D simulation tasks (which likely resemble familiar computer use applications) but neither succeeded in real-world tasks before exhausting our available credits, likely due to challenges in forceful interactions. We leave a controlled comparison to future work.

Task (score %)	SkillReasoner (Ours)			Play Data Baselines					Privileged Expert Demos		
	Low	Medium	High	GCIQL	HIQL	$\pi_{0.5}$	ACT	DP	$\pi_{0.5}$	ACT	DP
`narrow-passage`	81.55	91.09	**93.17**	13.58	14.16	3.04	4.15	3.86	27.21	23.34	36.89
`multi-room` (1 crossing)	82.46	92.26	**94.94**	1.33	1.73	1.92	4.43	2.17	3.70	6.36	15.73
`multi-room` (full)	72.02	81.90	**82.12**	0.82	1.11	1.15	0.87	0.73	6.51	3.81	8.13
`winding-maze` (1 crossing)	99.51	**100.00**	**100.00**	3.81	13.19	9.71	5.78	6.45	91.64	84.93	94.06
`winding-maze` (2 crossings)	99.16	99.02	**99.78**	1.21	4.92	3.35	2.05	2.32	77.73	76.67	87.38
`winding-maze` (full)	71.19	76.87	**87.44**	0.30	1.40	0.89	0.56	0.60	24.83	28.07	47.58

Table 1: **Normalized Task Score in Simulation.** Trained only on local play, SkillReasoner achieves unseen goals that require coordinated behaviors far longer than those seen in training, outperforming offline RL and behavior cloning baselines and even those trained on expert demonstrations. Low/Medium/High use search depths $1/2/4$, with planning rates of approximately 30/3/1 Hz in the `narrow-passage` timing benchmark. Details in Appendix F.

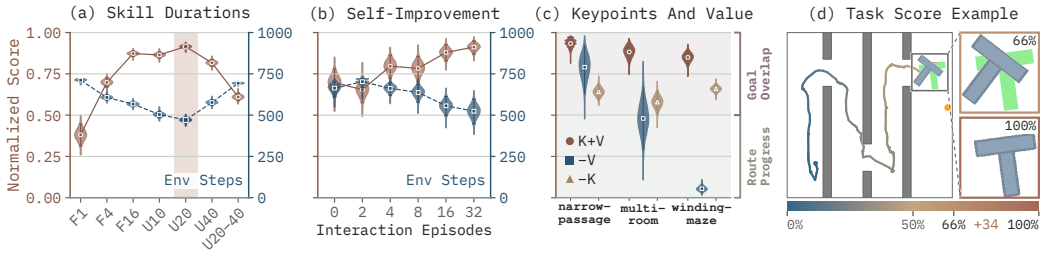

Figure 6: **Skill Duration Choice, Self-Improvement, Search Guidance, and Task Score Visualization.** (a) Varying horizons support reliable goal reaching with efficient execution. F: fixed; U: uniform. (b) Given fixed goal, SkillReasoner efficiently improves with few-shot interactions to succeed more often and faster. (c) Value (V) is crucial for route progress in search while keypoints (K) are critical for precise manipulation. (d) Reaching the goal region with misaligned orientation yields 66% score, and precise reorientation supplies the remaining 34 points. Plots (a,b,c) share the same left-sided y-axis showing normalized task score.

the robot, objects, and environment (Figure 4). Top-contact sliding transmits tangential forces and torsional moments through a finite frictional contact patch. Precise manipulation requires coordinating normal loading with planar motion to slide the object against table friction while controlling sticking and slipping at the gripper–object interface (Yi & Fazeli, 2023). The same end-effector path can therefore yield different object translations and rotations under different contact wrenches; for example, increased normal loading can reduce unwanted rotation. Side-contact pushing also couples translation and rotation: off-center contacts generate moments that depend on contact location and push direction. Compared with centered face pushes on regular rectangular blocks, our nonrectangular objects require shape-dependent contact placement and push directions to limit unwanted rotation, making precise goal alignment sensitive to both geometry and support friction. These effects motivate evaluating learned skills on tasks involving varied contact and force. We construct four tasks focusing on short-range alignment, longer-range transport, arranging an object into an initially blocked region, and packing several blocks into a region. We collect two hours of play data through teleoperation, instructing the operator to pursue self-selected goals using top-contact and side-contact behaviors (Lynch et al., 2020). No task-specific demonstrations are collected. All methods are trained on shared play data (with hindsight-relabeled goals for behavior cloning baselines). We report normalized task scores over ten trials per entry, using predetermined reset–goal pairs shared across methods. `short-arrange` and `long-arrange` scores equally weight direct L_2 distance progress and target coverage. `dense-pack` and `occluded-arrange` use target coverage alone to avoid favoring a particular manipulation strategy. Further details appear in Appendix E.

5 Experimental Findings

SkillReasoner composes skills from unstructured play to achieve unseen goals in both simulated and real-world contact-rich tasks (Table 1; Figure 5, bottom right). Goal-conditioned offline RL offers a framework for stitching data-supported behaviors into goal-reaching policies. However, in our tasks, GCIQL and HIQL make limited progress toward the target. A plausible explanation is difficult credit assignment across contact-mode transitions: extended free-space pusher

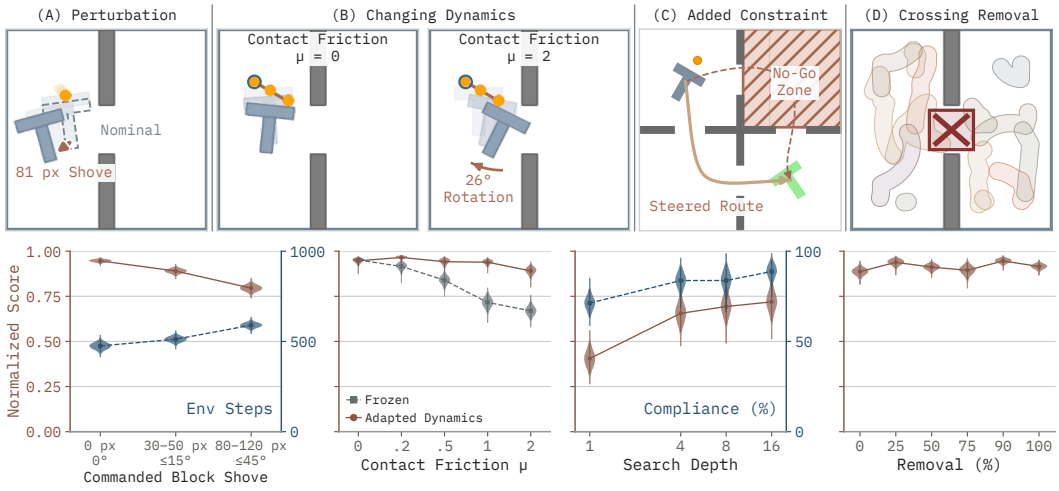

Figure 7: **Test-Time Adaptation and Composition.** (A) Randomly displace object during execution. (B) Changed contact friction at test time without training exposure or randomization. (C) Add a keep-out region while preserving an alternative route and enforce constraints through planning. (D) Remove complete crossings while retaining local play. The corresponding plots show that (A) replanning recovers from unseen shoves through additional corrective motion; (B) updating only dynamics restores performance under altered friction; (C) deeper search improves goal reaching and constraint compliance; and (D) retained local interactions support complete traversals without crossing-spanning examples.

motion provides little object-progress signal, so approximation error in bootstrapped Bellman targets can overwhelm small action-value differences (Park et al., 2023). Hindsight goal-conditioned behavior cloning baselines make almost no progress, likely because relabeling with achieved future poses supplies only within-episode goals, and these episodes are much shorter than the behaviors required at test time. Furthermore, play episodes may also push away from a test goal without recovering, effectively providing failure trajectories for that task. Privileged expert demonstrations yield substantially stronger policies, yet their normalized scores remain below those of our planner on every evaluated setting. These baselines also perform worse in `multi-room`, where alternative routes introduce additional solution multimodality. More expert demonstrations may help, but our aim is precisely generalization beyond training reset–goal distributions. On four tabletop alignment, transport, clearing, and packing tasks, SkillReasoner also achieves higher mean terminal geometric scores than hindsight-trained cloning when both use the same play data without task-specific demonstrations (Figure 5; Appendix E).

SkillReasoner performs progressively better with more test-time compute while supporting up to 30 Hz high-frequency replanning (Table 1). Across all environments in Table 1, increasing search depth (i.e., chained skills) from 1 to 4 improves task performance with the same fixed model. We hypothesize that deeper lookahead in search may balance errors in the learned cost-to-go estimate by evaluating more of the trajectory through locally accurate dynamics. The same skill space supports different planning budgets; the depth-one simulation configuration reaches roughly 30 Hz for planning computation at skill boundaries (Appendix F).

A varying-horizon skill space supports both reliable goal reaching and efficient execution (Figure 6(a)). Using `narrow-passage` play data, we compare fixed training horizons of 1, 4, and 16 steps with uniform ranges of 1–10, 1–20, 1–40, and 20–40. At fixed skill-search depth, duration ranges trade physical lookahead against fine adjustment. Varying horizons improve task performance and reduce mean execution steps when the range is chosen appropriately; Uniform 1–20 performs best here. Long-only variants often overshoot the goal, while Fixed-1 and Fixed-4 struggle to sustain progress, plausibly due to compounded decision errors. Varying horizons may let the planner match the task stage. Interestingly, extending the range to 1–40 hurts performance, plausibly because samples and model capacity are spread across more durations and the same search budget must cover a broader behavior space. Uniform 20–40 also performs poorly, consistent with the importance of short skills for precise goal alignment. Appendix Figure 8 shows the selected horizon distributions across environments.

Learned value guides long-horizon progress, while keypoints support precise goal alignment (Figure 6(c)). We ablate learned value and structured object guidance separately to examine their complementary roles. In `narrow-passage`, the relatively direct route often lets search reach the goal in imagination, despite demanding contact interactions. The longer routes in `multi-room` and `winding-maze` exceed this effective lookahead; simply searching deeper can exploit model errors, including predicted wall penetration. Learned cost-to-go guidance is therefore particularly valuable in `winding-maze`, where the value function guides the agent through successive passages even with approximation errors. However, value alone is insufficient across all three environments: without keypoint guidance, the planner can often transport the object into the goal region but struggles to reorient it into precise alignment, a central difficulty of the Push-T task (Chi et al., 2023). The distinction is between identifying promising regions and resolving the final pose: learned value supports the former, while explicit geometry supplies the precision needed for the latter.

SkillReasoner adapts to perturbations, changing dynamics, and test-time constraints while retaining learned skills (Figure 7). We test three scenarios unseen from training. *Perturbations.* Repeated random translations and rotations displace the block during execution, including near the goal. We find that replanning enables SkillReasoner to recover effectively, also shown by additional steps required to correct the setbacks. *Changing dynamics.* Adding pusher–block Coulomb friction alters tangential force at contact and the resulting object motion. Yet, the frozen model retains substantial progress; fine-tuning only skill dynamics on new interactions can further largely restore performance. *Test-time constraints.* We block one room in `multi-room` while leaving an alternative route. We steer search by rejecting candidates whose block footprint, decoded at predicted skill endpoints, violates the constraint. We find that deeper search improves both task performance and constraint compliance despite value guidance learned without the restriction, showing the benefit of separating behavior acquisition from test-time specification of novel objectives.

SkillReasoner can improve autonomously from efficient few-shot interactions at deployment (Figure 6(b)). We let the agent collect up to 32 episodes with a fixed reset to goal, updating both skill dynamics and the skill proposal between collection rounds. We find that on-policy experience efficiently adapts the dynamics to observed skill outcomes and refines the proposal through search distillation. This implements approximate model-based policy improvement (Sutton, 1990): we update the learned dynamics and proposal from experience, then replan to improve skill selection while keeping the skill decoder fixed. Using only 32 episodes, SkillReasoner approaches the maximum task score and progressively solves the task with fewer steps.

SkillReasoner composes local interactions into passage traversals even when passage-crossing data are removed (Figure 7(D)). To stress-test generalization beyond demonstrated routes, we progressively remove crossing-spanning training windows from `narrow-passage` while retaining local interactions around the passage. With all such windows removed, SkillReasoner still achieves high task performance comparable to full-data training (0.915 normalized score over 100 tasks). This suggests that continuity among the remaining local interactions can support a complete traversal without an intact crossing example.

6 Conclusion and Discussion

We introduced SkillReasoner to learn varying-horizon skills from offline interactions and compose them for new goals, separating behavior acquisition from task-specific planning. We argue that this separation is central to generalization in contact-rich manipulation: a robot should use what it has learned about physical interactions to *construct* a solution, rather than only learning from *given* ones (e.g., expert demos).

Limitations. Search depends on behavior coverage and model accuracy: it cannot create missing behaviors and can exploit prediction errors. New observations inform replanning at skill boundaries; execution within each skill remains open loop. The approach only generalizes to latent states familiar from training and thus its capability to handle novel environments is limited (e.g., unseen object types or layouts). The system is also not extended to support language-based task specification; our current goal features rely on structured object information or latent features encoded from goal images. A possible extension is to use a vision-language model to generate code specifying structured goals (Huang et al., 2025), or a text-to-image generative model to produce a goal image (Physical Intelligence et al., 2026).

AI USE STATEMENT

We used Codex and Claude Code for manuscript drafting and editing, literature discovery and analysis, code analysis, methodological discussion and hypothesis refinement, experiment implementation, result interpretation, and visualization and validation code. The authors take responsibility for the final content.

REPRODUCIBILITY STATEMENT

The main text and appendix describe the method, implementation, and evaluation protocol. We will open-source our code and publicly release model checkpoints, datasets, and experimental artifacts to support reproducibility and further research.

REFERENCES

David Abel, Nate Umbanhowar, Khimya Khetarpal, Dilip Arumugam, Doina Precup, and Michael L. Littman. Value preserving state-action abstractions. In *International Conference on Artificial Intelligence and Statistics*, pp. 1639–1650, 2020. URL https://proceedings.mlr.press/v108/abel20a.html.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. URL https://arxiv.org/abs/2204.01691.

Anurag Ajay, Aviral Kumar, Pulkit Agrawal, Sergey Levine, and Ofir Nachum. OPAL: Offline primitive discovery for accelerating offline reinforcement learning. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.13611.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. URL https://arxiv.org/abs/2506.09985.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *AAAI Conference on Artificial Intelligence*, 2017. URL https://ojs.aaai.org/index.php/AAAI/article/view/10916.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *Conference on Robot Learning*, 2025. URL https://proceedings.mlr.press/v305/black25a.html.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.026. URL https://roboticsproceedings.org/rss19/p026.html.

Thomas G. Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research*, 13:227–303, 2000. URL https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume13/dietterich00a.pdf.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1802.06070.

Julian Förster, Lionel Ott, Juan Nieto, Nicholas Lawrance, Roland Siegwart, and Jen Jen Chung. Automatic extension of a symbolic mobile manipulation skill set. *Robotics and Autonomous Systems*, 165:104428, 2023. doi: 10.1016/j.robot.2023.104428. URL https://doi.org/10.1016/j.robot.2023.104428.

Caelan Reed Garrett, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. PDDLStream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, 2020. doi: 10.1609/icaps.v30i1.6739. URL https://ojs.aaai.org/index.php/ICAPS/article/view/6739.

Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:265–293, 2021. doi: 10.1146/annurev-control-091420-084139. URL https://www.annualreviews.org/content/journals/10.1146/annurev-control-091420-084139.

Gemini Robotics Team et al. Gemini Robotics: Bringing AI into the physical world. *arXiv preprint arXiv:2503.20020*, 2025. doi: 10.48550/arXiv.2503.20020. URL https://arxiv.org/abs/2503.20020.

Bernhard Paus Graesdal, Shao Yuan Chew Chia, Tobia Marcucci, Savva Morozov, Alexandre Amice, Pablo Parrilo, and Russ Tedrake. Towards tight convex relaxations for contact-rich manipulation. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.132. URL https://www.roboticsproceedings.org/rss20/p132.html.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/pdf?id=FzpfPa6unv.

Jean Harb, Pierre-Luc Bacon, Martin Klissarov, and Doina Precup. When waiting is not an option: Learning options with a deliberation cost. In *AAAI Conference on Artificial Intelligence*, 2018. URL https://ojs.aaai.org/index.php/AAAI/article/view/11831.

Kris Hauser and Jean-Claude Latombe. Multi-modal motion planning in non-expansive spaces. In *Workshop on the Algorithmic Foundations of Robotics*, 2008. URL https://hauserlab.web.illinois.edu/papers/wafr08-multimodal.pdf.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9118–9147. PMLR, 2022. URL https://proceedings.mlr.press/v162/huang22a.html.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. VoxPoser: Composable 3D value maps for robotic manipulation with language models. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 540–562, 2023a. URL https://proceedings.mlr.press/v229/huang23b.html.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Tomas Jackson, Noah Brown, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning through planning with language models. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1769–1782, 2023b. URL https://proceedings.mlr.press/v205/huang23c.html.

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. ReKep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4573–4602. PMLR, 2025. URL https://proceedings.mlr.press/v270/huang25g.html.

Wenlong Huang, Yu-Wei Chao, Arsalan Mousavian, Ming-Yu Liu, Dieter Fox, Kaichun Mo, and Li Fei-Fei. PointWorld: Scaling 3D world models for in-the-wild robotic manipulation. *arXiv preprint arXiv:2601.03782*, 2026. URL https://arxiv.org/abs/2601.03782.

Yiding Jiang, Evan Zheran Liu, Benjamin Eysenbach, J. Zico Kolter, and Chelsea Finn. Learning options via compression. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/8567a53e58a9fa4823af356c76ed943c-Abstract-Conference.html.

Yuu Jinnai, David Abel, D. Ellis Hershkowitz, Michael L. Littman, and George Konidaris. Finding options that minimize planning time. In *International Conference on Machine Learning*, pp. 3120–3129, 2019. URL https://proceedings.mlr.press/v97/jinnai19a.html.

Leslie Pack Kaelbling. Hierarchical learning in stochastic domains: Preliminary results. In *Proceedings of the Tenth International Conference on Machine Learning*, pp. 167–173, 1993. doi: 10.1016/B978-1-55860-307-3.50028-9. URL https://doi.org/10.1016/B978-1-55860-307-3.50028-9.

Leslie Pack Kaelbling and Tomás Lozano-Pérez. Hierarchical task and motion planning in the now. In *IEEE International Conference on Robotics and Automation*, 2011. URL https://people.csail.mit.edu/lpk/papers/hpnICRA11Final.pdf.

Lydia E. Kavraki, Petr Svestka, Jean-Claude Latombe, and Mark H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996. doi: 10.1109/70.508439. URL https://kavrakilab.rice.edu/publications/kavraki-svestka1996probabilistic-roadmaps-for.html.

George Konidaris and Andrew G. Barto. Skill discovery in continuous reinforcement learning domains using skill chaining. In *Advances in Neural Information Processing Systems*, 2009. URL https://papers.nips.cc/paper_files/paper/2009/hash/e0cf1f47118daebc5b16269099ad7347-Abstract.html.

George Konidaris, Scott Kuindersma, Roderic Grupen, and Andrew Barto. Robot learning from demonstration by constructing skill trees. *The International Journal of Robotics Research*, 31 (3):360–375, 2012. doi: 10.1177/0278364911428653. URL https://doi.org/10.1177/0278364911428653.

George Konidaris, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. From skills to symbols: Learning symbolic representations for abstract high-level planning. *Journal of Artificial Intelligence Research*, 61:215–289, 2018. URL https://cs.brown.edu/people/gdk/pubs/orig_sym_jair.pdf.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2110.06169.

Nishanth Kumar, Tom Silver, Willie McClinton, Linfeng Zhao, Stephen Proulx, Tomás Lozano-Pérez, Leslie Pack Kaelbling, and Jennifer Barry. Practice makes perfect: Planning to learn skill parameter policies. In *Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.040. URL https://www.roboticsproceedings.org/rss20/p040.html.

Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, Winson Han, Wilbert Pumacay, Angelica Wu, Rose Hendrix, Karen Farley, Eli VanderBilt, Ali Farhadi, Dieter Fox, and Ranjay Krishna. MolmoAct: Action reasoning models that can reason in space. *arXiv preprint arXiv:2508.07917*, 2025. URL https://arxiv.org/abs/2508.07917.

Kevin Lin, Christopher Agia, Toki Migimatsu, Marco Pavone, and Jeannette Bohg. Text2Motion: From natural language instructions to feasible plans. *Autonomous Robots*, 47(8):1345–1365, 2023. doi: 10.1007/s10514-023-10131-7. URL https://link.springer.com/article/10.1007/s10514-023-10131-7.

Chaoqi Liu, Xiaoshen Han, Jiawei Gao, Yue Zhao, Haonan Chen, and Yilun Du. OAT: Ordered action tokenization. In *Proceedings of Robotics: Science and Systems*, 2026. doi: 10.15607/RSS.2026.XXII.075. URL https://www.roboticsproceedings.org/rss22/p075.html.

Corey Lynch and Pierre Sermanet. Language Conditioned Imitation Learning Over Unstructured Data. In *Proceedings of Robotics: Science and Systems*, 2021. doi: 10.15607/RSS.2021.XVII.047. URL https://roboticsproceedings.org/rss17/p047.html.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1113–1132, 2020. URL https://proceedings.mlr.press/v100/lynch20a.html.

Timothy A. Mann, Shie Mannor, and Doina Precup. Approximate value iteration with temporally extended actions. *Journal of Artificial Intelligence Research*, 53:375–438, 2015. URL https://doi.org/10.1613/jair.4676.

Jiayuan Mao, Tomás Lozano-Pérez, Joshua B. Tenenbaum, and Leslie Pack Kaelbling. PDSketch: Integrated domain programming, learning, and planning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 36972–36984, 2022. doi: 10.52202/068431-2679. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/efe36e55d80a94d1726f660b8d237a0f-Abstract-Conference.html.

Amy McGovern and Andrew G. Barto. Automatic discovery of subgoals in reinforcement learning using diverse density. In *Proceedings of the 18th International Conference on Machine Learning*, pp. 361–368, 2001. URL https://mcgovern-fagg.org/amy_html/pubs/mcgovern_barto_icml2001.pdf.

Ishai Menache, Shie Mannor, and Nahum Shimkin. Q-Cut—dynamic discovery of sub-goals in reinforcement learning. In *Machine Learning: ECML 2002*, volume 2430 of *Lecture Notes in Computer Science*, pp. 295–306, 2002. doi: 10.1007/3-540-36755-1_25. URL https://doi.org/10.1007/3-540-36755-1_25.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/e2dd53601de57c773343a7cdf09fae1c-Abstract-Conference.html.

Atharva Mete, Haotian Xue, Albert Wilcox, Yongxin Chen, and Animesh Garg. QueST: Self-supervised skill abstractions for learning continuous control. In *Advances in Neural Information Processing Systems*, volume 37, pp. 4062–4089, 2024. doi: 10.52202/079017-0133. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/076c3e48fa502c660902105965fdd9f6-Abstract-Conference.html.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1802.05957.

Ofir Nachum, Shixiang Gu, Honglak Lee, and Sergey Levine. Near-optimal representation learning for hierarchical reinforcement learning. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1810.01257.

Ashvin Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/7ec69dd44416c46745f6edd947b470cd-Abstract.html.

Neil Nie, Wenlong Huang, Jiayuan Mao, Li Fei-Fei, Weiyu Liu, and Jiajun Wu. Learning composable skills by discovering spatial and temporal structure with foundation models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026. URL https://icra-stack.github.io/.

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*, 2023. URL https://arxiv.org/abs/2307.11949.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *International Conference on Learning Representations*, 2025a. URL https://ogbench.org.

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*, volume 38, pp. 9258–9297, 2025b. doi: 10.52202/085713-0284. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/0c66099070429b9b49ad8bb19c740c22-Paper-Conference.pdf.

Karl Pertsch, Youngwoon Lee, and Joseph J. Lim. Accelerating reinforcement learning with learned skill priors. In *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pp. 188–204, 2021. URL https://proceedings.mlr.press/v155/pertsch21a.html.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.012. URL https://www.roboticsproceedings.org/rss21/p012.html.

Physical Intelligence et al. $\pi_{0.7}$: A steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026. doi: 10.48550/arXiv.2604.15483. URL https://arxiv.org/abs/2604.15483.

Doina Precup. *Temporal Abstraction in Reinforcement Learning*. PhD thesis, University of Massachusetts Amherst, 2000.

Seungeun Rho, Laura Smith, Tianyu Li, Sergey Levine, Xue Bin Peng, and Sehoon Ha. Language guided skill discovery. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/pdf?id=i3e92uSZCp.

Erick Rosete-Beas, Oier Mees, Gabriel Kalweit, Joschka Boedecker, and Wolfram Burgard. Latent plans for task-agnostic offline reinforcement learning. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1838–1849, 2023. URL https://proceedings.mlr.press/v205/rosete-beas23a.html.

Yifei Simon Shao, Yuchen Zheng, Sunan Sun, Pratik Chaudhari, Vijay Kumar, and Nadia Figueroa. SymSkill: Symbol and skill co-invention for data-efficient and reactive long-horizon manipulation. In *IEEE International Conference on Robotics and Automation*, 2026. URL https://symskill.github.io/.

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1907.01657.

Lucy Xiaoyang Shi, Joseph J Lim, and Youngwoon Lee. Skill-based model-based reinforcement learning. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 2262–2272, 2023. URL https://proceedings.mlr.press/v205/shi23a.html.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go

with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. URL https://doi.org/10.1038/nature16961.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025. URL https://arxiv.org/abs/2508.10104.

Alec Solway, Carlos Diuk, Natalia Córdova, Debbie Yee, Andrew G. Barto, Yael Niv, and Matthew M. Botvinick. Optimal behavioral hierarchy. *PLoS Computational Biology*, 10(8): e1003779, 2014. URL https://doi.org/10.1371/journal.pcbi.1003779.

Richard S. Sutton. Integrated modeling and control based on reinforcement learning and dynamic programming. In *Advances in Neural Information Processing Systems*, volume 3, pp. 471–478. Morgan-Kaufmann, 1990. URL https://proceedings.neurips.cc/paper_files/paper/1990/file/d9fc5b73a8d78fad3d6dffe419384e70-Paper.pdf.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1–2):181–211, 1999. URL https://doi.org/10.1016/S0004-3702(99)00052-1.

Momchil S. Tomov, Samyukta Yagati, Agni Kumar, Wanqian Yang, and Samuel J. Gershman. Discovery of hierarchical representations for efficient planning. *PLoS Computational Biology*, 16 (4):e1007594, 2020. doi: 10.1371/journal.pcbi.1007594. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1007594.

TRI LBM Team et al. A careful examination of large behavior models for multitask dexterous manipulation. *Science Robotics*, 11(113):eaea6201, 2026. doi: 10.1126/scirobotics.aea6201. URL https://arxiv.org/abs/2507.05331.

Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. FoundationPose: Unified 6D pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://nvlabs.github.io/FoundationPose/.

Bowen Wen, Shaurya Dewan, and Stan Birchfield. Fast-FoundationStereo: Real-time zero-shot stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://nvlabs.github.io/Fast-FoundationStereo/.

Kevin Xie, Homanga Bharadhwaj, Danijar Hafner, Animesh Garg, and Florian Shkurti. Latent skill planning for exploration and transfer. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2011.13897.

Xili Yi and Nima Fazeli. Precise object sliding with top contact via asymmetric dual limit surfaces. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.045. URL https://www.roboticsproceedings.org/rss19/p045.html.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 3157–3181. PMLR, 2025. URL https://proceedings.mlr.press/v270/zawalski25a.html.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.016. URL https://roboticsproceedings.org/rss19/p016.html.

Yifeng Zhu, Peter Stone, and Yuke Zhu. Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation. *IEEE Robotics and Automation Letters*, 7(2):4126–4133, 2022. doi: 10.1109/LRA.2022.3146589. URL https://www.cs.utexas.edu/~pstone/Papers/bib2html/b2hd-ral2022-zhu.html.

A Extended Related Work

Temporal Abstraction. Options formalize temporally extended behavior (Sutton et al., 1999; Precup, 2000), often with learned termination (Bacon et al., 2017; Harb et al., 2018). Our varying-horizon skills build on this flexibility to balance planning efficiency (Mann et al., 2015; Jinnai et al., 2019) with preservation of high-value behavior under abstraction (Abel et al., 2020; Nachum et al., 2019), a tension also reflected in accounts of human hierarchy (Solway et al., 2014; Tomov et al., 2020). Where skill discovery objectives often emphasize diversity (Eysenbach et al., 2019), diverse, predictable effects (Sharma et al., 2020), or language-grounded semantic diversity (Rho et al., 2025), we seek skills whose actions, durations, and predicted effects support contact-rich manipulation planning. While online interactions can extend symbolic skill sets (Förster et al., 2023) or improve skill-parameter policies through planned practice (Kumar et al., 2024), we instead learn the skill vocabulary from offline data for scalable training. Unlike action chunks commonly used in imitation learning (Zhao et al., 2023) or tokenizers trained on action sequences alone (Pertsch et al., 2025; Liu et al., 2026; Mete et al., 2024), we compress varying-horizon action sequences by also conditioning on observations, enabling flexible compositions of motions for highly constrained tasks.

Learning from Unstructured Play. Play data decouples behavior acquisition from downstream tasks, supporting learning from unstructured data (Lynch et al., 2020), with possibly additional language annotations (Lynch & Sermanet, 2021). Structure extracted from unsegmented demonstrations (Zhu et al., 2022) or encouraged through compression (Jiang et al., 2022) can guide exploration (Pertsch et al., 2021) or offline policy optimization (Ajay et al., 2021), where horizon reduction can improve scalability (Park et al., 2025b). Closely related work composes learned skills from play through a goal-conditioned high-level policy learned with offline RL (Rosete-Beas et al., 2023); our formulation instead uses search for skill composition, which enables generalization to new constraints specified at test time. Joint skill and dynamics learning also supports downstream model-based RL (Shi et al., 2023). We instead focus on scalable learning from offline data and its application to contact-rich manipulation tasks.

Test-Time Reasoning in Manipulation. Task and motion planning performs test-time search by coupling high-level discrete decisions to low-level continuous feasibility (Kaelbling & Lozano-Pérez, 2011; Garrett et al., 2021; 2020; Mao et al., 2022), including symbols learned from skill effects (Konidaris et al., 2018) or jointly with skills (Shao et al., 2026); our formulation focuses on learning skills through compression from unstructured interactions. Language-based hierarchical systems often perform test-time reasoning in the space of language or code, often done by LLMs or VLMs (Huang et al., 2023b; Zawalski et al., 2025), with grounding through skill feasibility (Ahn et al., 2022), geometric checks (Lin et al., 2023), spatial objectives (Huang et al., 2023a), or trajectory traces (Lee et al., 2025). We target contact-level test-time reasoning beneath the linguistic abstraction; related methods use physics-based planning (Graesdal et al., 2024) or learned world models, either to train an actor in imagination (Hafner et al., 2020) or to plan online (Hansen et al., 2024; Assran et al., 2025; Huang et al., 2026). Planning over learned skills and their dynamics has been studied in prior work (Sharma et al., 2020; Xie et al., 2021; Shi et al., 2023; Nie et al., 2026). In contrast, we focus on skill-level reasoning of varying temporal granularity, composing extended motions and short adjustments within the same learned skill space.

B PLANNING PSEUDOCODE

Algorithm 1 details the latent skill search and closed-loop execution described in §3.2.

Algorithm 1 Latent Skill Planning (Inference)

Input: state encoder E, skill proposal P, skill dynamics F, value function V, skill decoder D, grounding head G
Input: goal y_g with encoded goal examples for V
Input: high-level policy π_{hi}, budget $\mathcal{B} = (L, B, N)$, temperature τ

Main Deployment Loop

while goal not reached **do**
  $z \leftarrow E(h_t)$ // encode observed history
  $u_{1:m} \leftarrow \pi_{\text{hi}}(z, y_g; \mathcal{B}, \tau)$
  if $m = 0$ **then stop**
  $(\hat{a}_{t:t+H-1}, p_D) \leftarrow D(z, u_1)$; $\hat{d} \leftarrow \arg\max_d p_D(d \mid z, u_1)$
  Execute $\hat{a}_{t:t+\hat{d}-1}$; update history and $t \leftarrow t + \hat{d}$
end while

Search-Based High-Level Policy

procedure $\pi_{\text{hi}}(z, y_g; \mathcal{B}, \tau)$
  $\mathcal{Z} \leftarrow \{(z, \varnothing, 0)\}$; $\mathcal{V} \leftarrow \varnothing$ // state, skill prefix, path risk
  for $\ell = 1, \ldots, L$ **do** // search depth counts skills
    $\mathcal{X} \leftarrow \varnothing$ // candidate plans at this depth
    for all $(z', s, r) \in \mathcal{Z}$ **do**
      $u^{(1:N)} \sim P_\tau(\cdot \mid z')$
      $z^{(n)} \leftarrow F(z', u^{(n)})$ for all n // predict endpoints in parallel
      $r^{(n)} \leftarrow r + \rho(\nu_y(z^{(n)}))$ for all n // add endpoint risk to path risk
      $\mathcal{X} \leftarrow \mathcal{X} \cup \{(z^{(n)}, s \,\|\, u^{(n)}, r^{(n)})\}_{n=1}^{N}$
    end for
    $\mathcal{Z} \leftarrow$ up to B valid, distinct candidates in $\mathcal{X}$ with lowest J_ℓ
    $\mathcal{V} \leftarrow \mathcal{V} \cup \mathcal{Z}$ // shorter plans remain eligible
    if goal reached **then return** lowest-cost successful prefix in $\mathcal{Z}$
    if $\mathcal{Z} = \varnothing$ **then break**
  end for
  return the lowest-cost prefix in $\mathcal{V}$, or $\varnothing$ if empty
end procedure

C Model and Training Details

We first learn the state and skill representations, then freeze them to train the proposal and dynamics (§3.3). Tables 2 and 3 summarize the planar model. We define the losses and goal-conditioned value targets below.

Table 2: **Model architecture and interfaces.** Widths denote feature dimensions.

Component	Specification
Visual backbone	Shared DINOv3 (Siméoni et al., 2025); finetuned in stage 1, frozen in stage 2.
State encoder E	Four observations and three intervening actions ($c = 3$); 512-dimensional latent state.
Sequence embeddings	Modality and relative-time embeddings.
Skill encoder Q	Attention-based; reads observations and actions through the sampled endpoint.
Skill representation	$K = 4$ FSQ tokens; levels $(8, 8, 4)$, or 256 values per token (Mentzer et al., 2024).
Skill decoder D	Block-causal Transformer; predicts actions and skill horizon.
Dynamics F	Six residual blocks; width 768, feed-forward width 1536; rank-192 state–skill modulation.
Dynamics skill mixer	Two layers; width 256, four attention heads, feed-forward width 512.
Proposal P	Autoregressive; three residual layers of width 256, feed-forward width 512; low-rank state conditioning.
Grounding head G	Nine object keypoints and pusher position, with uncertainties: 20 coordinates per history position. Object poses are recovered from keypoints for planning.
Grounding and goals	Simulator states supply training targets, never visual-encoder inputs. Evaluation goals use physical coordinates.

Table 3: **Sampling, objectives, and optimization.** Stage 1 learns representations; stage 2 learns proposal and dynamics.

Setting	Value or choice
Window sampling	Uniform horizon in 1–20 control steps, then uniformly sampled eligible episode and start index.
Stage-1 loss weights	$(\lambda_a, \lambda_d, \lambda_g) = (1.5, 0.75, 0.1875)$.
Loss reductions	Action: mean over valid coordinates; horizon cross-entropy: mean over examples; proposal cross-entropy: mean over token positions.
Grounding loss	Separate means for object and pusher coordinates, weighted $2/3$ and $1/3$; supervise all eligible observed histories in stage 1.
Auxiliary dynamics target	Sampled skill horizon; execution uses the frozen decoder's horizon prediction.
Optimizers	Muon for hidden matrices, including DINO attention matrices in stage 1; AdamW for remaining parameters.
Stage-2 batch size	1024.
Stage-2 learning rate	5.25×10^{-4}.
Stage-2 warmup	Linear, over the first 32,000 sampled windows.
Dynamics stabilization	Spectral cap $\kappa = 1.5$ per linear map; largest singular value estimated by power iteration.

We stabilize repeated dynamics predictions by applying

$$\widetilde{W} = \frac{W}{\max(1, \widehat{\sigma}_{\max}(W)/\kappa)}. \tag{2}$$

This limits amplification by individual matrices; it does not bound the complete nonlinear transition.

For context $h_t = (o_{t-c:t}, a_{t-c:t-1})$ and a duration-d window $w_{t:t+d} = (o_{t-c:t+d}, a_{t-c:t+d-1})$, we use the following objectives. Table 3 summarizes loss reductions and stage-1 weights; value targets are defined in Appendix C.1.

The grounding head predicts a mean $\hat{y}$ and positive coordinatewise variance $\nu_y(z)$:

$$\mathcal{L}^{\text{obs}}_{\text{ground}}(z, y) = \frac{1}{2} \sum_k \left[\frac{(y_k - \hat{y}_k)^2}{\nu_{y,k}(z)} + \log \nu_{y,k}(z) \right]. \tag{3}$$

For a duration-d window, the skill code contains K discrete tokens,

$$u^\star = Q(w_{t:t+d}) = \big(u^1, \dots, u^K\big), \qquad u^k \in \mathcal{C}, \qquad \mathcal{U} = \mathcal{C}^K. \tag{4}$$

The decoder is trained with heteroscedastic action regression and duration classification:

$$\mathcal{L}_{\text{act}} = \frac{1}{2} \sum_{j=0}^{d-1} \sum_{k} \left[\frac{(a_{t+j,k} - \hat{a}_{j,k})^2}{\nu_{a,j,k}} + \log \nu_{a,j,k} \right], \quad \mathcal{L}_{\text{dur}} = -\log p_D(d \mid z_t, u^\star). \tag{5}$$

The proposal factorizes the token distribution and is trained with stop-gradient targets:

$$p_P(u \mid z) = \prod_{k=1}^{K} p_P(u^k \mid z, u^{<k}), \qquad \mathcal{L}_{\text{proposal}} = -\sum_{k=1}^{K} \log p_P(u^{\star,k} \mid \text{sg}(z_t), u^{\star,<k}). \tag{6}$$

The dynamics objective predicts the encoded endpoint and its grounded features:

$$\mathcal{L}_{\text{dyn}} = \frac{1}{n_z} \|\hat{z}_{t+d} - z_{t+d}\|^2, \qquad \mathcal{L}^{\text{pred}}_{\text{ground}} = \mathcal{L}^{\text{obs}}_{\text{ground}}(\hat{z}_{t+d}, y_{t+d}). \tag{7}$$

Finally, the value function uses expectile regression,

$$\mathcal{L}_{\text{value}} = \mathbb{E}\Big[|\eta - \mathbf{1}[\hat{v} < V(z, z_g)]| \, \big(\hat{v} - V(z, z_g)\big)^2 \Big], \tag{8}$$

where $\hat{v}$ is the duration-aware target and $\eta \in (1/2, 1)$ is the expectile level. The two training stages combine these terms as

$$\mathcal{L}_1 = \lambda_a \mathcal{L}_{\text{act}} + \lambda_d \mathcal{L}_{\text{dur}} + \lambda_g \mathcal{L}^{\text{obs}}_{\text{ground}}, \tag{9}$$

$$\mathcal{L}_2 = \lambda_u \mathcal{L}_{\text{proposal}} + \lambda_z \mathcal{L}_{\text{dyn}} + \lambda_y \mathcal{L}^{\text{pred}}_{\text{ground}} + \lambda_v \mathcal{L}_{\text{value}}. \tag{10}$$

C.1 Goal-Conditioned Value Learning

We train the value function V on frozen state and goal encodings during the second training stage. We sample hindsight goals from recorded histories and include pairs in which the current history is itself the goal, anchoring its remaining cost at zero.

For a transition (z, z', d), let $b \in \{0, 1\}$ indicate whether the endpoint reaches the relabeled goal. We assign a cost to each control step, so a skill spanning d steps receives reward $r_d = -s^{-1} \sum_{j=0}^{d-1} \gamma^j$, where $s > 0$ rescales the cost and $\gamma \in (0, 1]$ is the discount. The value target is

$$\hat{v} = \begin{cases} 0, & \text{if the current history is the goal,} \\ \text{clip}\big(r_d + \gamma^d (1 - b) \bar{V}(z', z_g),\, -C,\, 0\big), & \text{otherwise.} \end{cases} \tag{11}$$

Here $\bar{V}$ is an exponential-moving-average target value function, and $C > 0$ bounds the magnitude of the target. Reaching the goal removes the continuation cost but retains the cost of the executed actions. With $\gamma = 1$, the reward reduces to $-d/s$. We hold the target fixed during each update and minimize the expectile loss in Equation (8), which favors better goal-reaching continuations in the data (Kostrikov et al., 2022; Park et al., 2023).

When several goal examples represent y_g, we average their predicted values, clip the result to $[-C, 0]$, and negate it to obtain the search heuristic c_{h}. The geometric goal cost remains useful for precise completion, while the value estimates the cost of reaching the goal.

C.2 SEARCH CONFIGURATION

Table 4: **Simulation search configuration.** Settings are shared across environments; Low, Medium, and High change only search depth.

Setting	Value
Beam width B	160
Proposals per parent N	384
Sampling temperature τ	1.3
Value weight	1
Risk weight	0.005
Search depth (skills)	1 / 2 / 4 (Low / Medium / High)
Accumulated duration cost	None
Duplicate-state thresholds	Object / pusher position: 8 units; orientation: 0.25 rad
Selection and execution	Select the lowest-cost successful prefix on early stopping, otherwise the lowest-cost noninitial retained prefix; execute its first skill, then replan.
Early stopping in search	Enabled at predicted goal satisfaction

D Composition Across Temporal Scales

We show selected skill durations and execution timelines for 120 depth-four rollouts across three environments (20 tasks and two value-head seeds per environment), using geometric and learned value guidance. We hypothesize that frequent one-step choices provide fine corrections or near-stationary actions when longer candidates offer little progress, allowing the planner to avoid committing to an unhelpful extended motion.

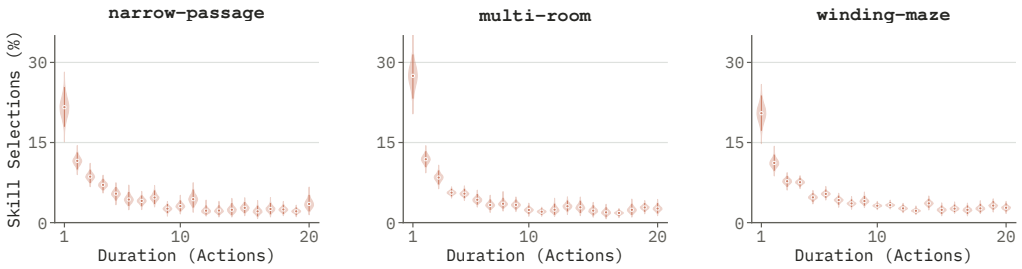

Figure 8: **Selected Skill Duration Distributions.** Percentage of decisions at each planned horizon, including all outcomes. Frequencies are normalized per rollout, then averaged over seeds and tasks. Circles show means; violins show task-bootstrap means; whiskers mark 95% intervals (5,000 resamples). Each skill counts once, regardless of its execution length.

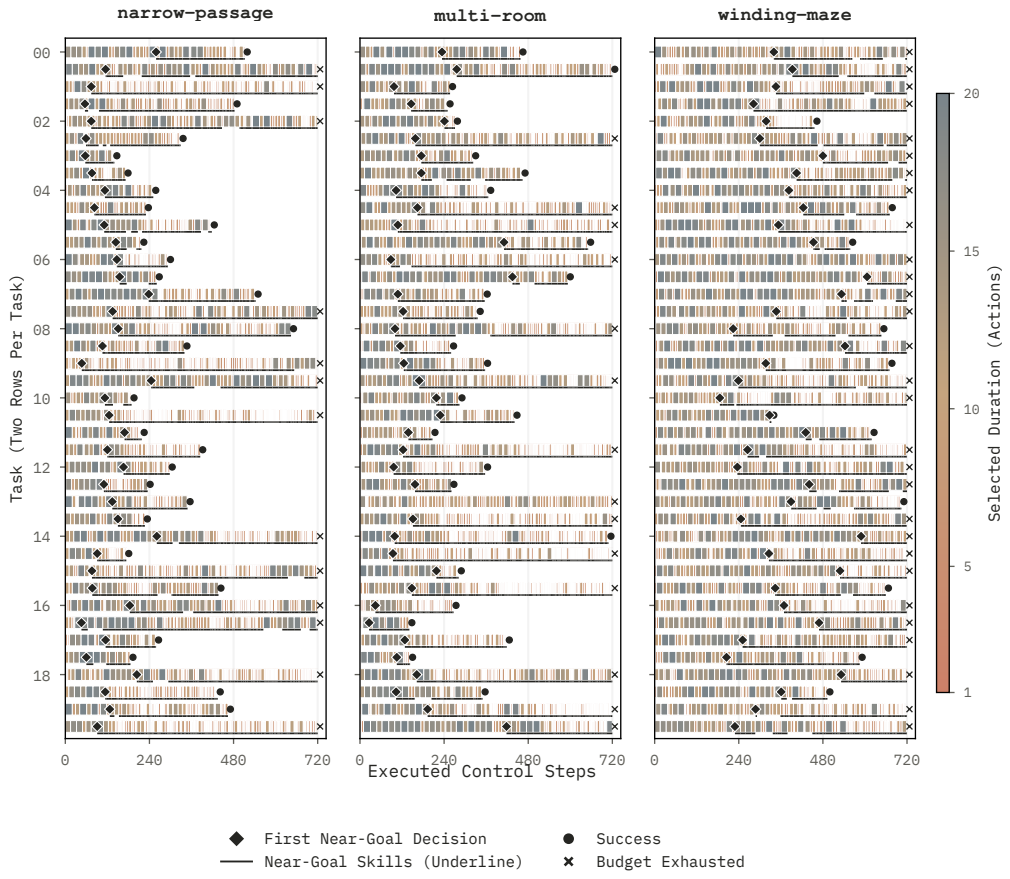

Figure 9: **Executed Skill Composition.** Each row is a rollout; segment width shows executed duration and color shows planned duration. Diamonds mark the first near-goal decision; underlines mark skills starting within 60 px of the goal centroid or at least 0.5 overlap. Endpoints indicate success or the execution limit.

E Real-World Setup and Evaluation Protocol

Platform and data collection. We use the right arm of a Marvin M6 with its gripper closed for top and side contacts. We use a wrist camera and two cameras viewing the scene, as shown in Figure 4. We record timestamped joint states, tool poses, wrist wrenches, and commanded targets. We collect play through VR teleoperation, mapping end-effector targets into the robot base frame and executing them through differential inverse kinematics and joint impedance control. Action labels use commanded tool poses; observations include measured robot state and wrist wrench.

Offline geometric annotation. We estimate stereo depth with Fast-FoundationStereo (Wen et al., 2026) and track object poses with FoundationPose (Wen et al., 2024). We use these offline pose estimates only to supervise the object-keypoint features that support goal specification, masking missing observations. At deployment, G predicts these features from camera observations and imagined latent states for comparison with the specified goal; neither the policy nor search receives tracked object poses or requires an online pose tracker.

Evaluation. We evaluate each method on ten predetermined reset–goal pairs per task, shared across methods. We restore each scene using reference images and poses before execution. The four tasks are:

- `short-arrange`: position and orient a block within a nearby object-shaped target.
- `long-arrange`: transport and align a block with a more distant object-shaped target.
- `occluded-arrange`: clear obstructing blocks and place the yellow triangle in its target, allowing any clearing order.
- `dense-pack`: arrange several blocks within a rectangular region, without prescribing their order or individual final poses.

Normalized task scores. Let B_T be the final object footprint and G the target region. We measure coverage as $o_T = |B_T \cap G|/|G|$, rather than intersection-over-union. For short- and long-range rearrangement, distance progress is $p_T = \mathrm{clip}(1 - d_T/d_0, 0, 1)$, where d_t is the object–goal centroid distance and $d_0 > 0$. Scores are

$$S_{\text{short}} = S_{\text{long}} = \tfrac{1}{2}p_T + \tfrac{1}{2}o_T, \qquad S_{\text{occluded}} = o_T^{\text{yellow}}, \qquad S_{\text{pack}} = \frac{|G \cap (\bigcup_j B_{j,T})|}{|G|}. \tag{12}$$

We use direct L_2 distance because the tabletop has no fixed walls requiring nominal-route distances as in simulation. Occluded rearrangement and dense packing use coverage alone to avoid favoring a particular manipulation strategy; the footprint union prevents double-counting packed objects. All scores use the terminal state, including early stops, and lie in $[0, 1]$.

Following the violin presentation of TRI LBM Team et al. (2026), we plot mean-score distributions from 20,000 paired trial-bootstrap resamples, with dots for observed means and whiskers for 95% intervals.

F SINGLE-ENVIRONMENT PLANNING LATENCY

We measure synchronous observation-to-action computation in `narrow-passage`, including visual encoding, search, and action decoding, but excluding startup, rendering, and action execution. We use an RTX 4090 and Threadripper PRO 5975WX (four CPU threads), PyTorch 2.7.1/CUDA 12.6, and FP32 with TF32 disabled. Inference uses CUDA graphs and cached weights.

We use the `narrow-passage` model and the shared search budgets in Table 4. For consistent timing comparisons, the benchmark disables early stopping and uses a fixed duration cost of 0.003 per control step. We time three consecutive decisions at each of five starts per depth after warmup, synchronizing CUDA for wall-clock measurements. Depth-one planning reaches approximately 30 Hz; deeper search trades speed for lookahead. Planning occurs at skill boundaries, separately from low-level control.

Table 5: **Planning latency in** `narrow-passage`**.** Median / empirical 95th percentile in milliseconds over 15 decisions per setting.

Environment	**Depth 1**	**Depth 2**	**Depth 4**
`narrow-passage`	34.2 / 38.2	344.8 / 351.4	979.2 / 1013.8

G Full Simulation Results

Table 6 expands the terminal task scores in Table 1 with success rates and execution lengths. Success requires 95% object–goal overlap; steps include unsuccessful trials, which run to the 720-step limit. For SkillReasoner, the two runs use separate value heads with the same frozen skill model; baseline runs use separate training seeds. Reported means average model runs within each task and then across tasks.

Table 6: **Task completion and execution behind Table 1.** Score and success are percentages; steps are mean environment controls. Play and expert indicate the training data.

Method	Data	Score	Success	Steps
`narrow-passage`				
SkillReasoner (Low)	Play	81.55	39.0	624.5
SkillReasoner (Medium)	Play	91.09	73.0	458.5
SkillReasoner (High)	Play	93.17	66.5	455.4
GCIQL	Play	13.58	0.0	720.0
HIQL	Play	14.16	0.0	720.0
$\pi_{0.5}$	Play	3.04	0.0	720.0
ACT	Play	4.15	0.7	716.5
DP	Play	3.86	0.3	718.8
$\pi_{0.5}$	Expert	27.21	0.0	720.0
ACT	Expert	23.34	0.0	720.0
DP	Expert	36.89	2.0	712.8
`multi-room` **(1 crossing)**				
SkillReasoner (Low)	Play	82.46	36.5	631.8
SkillReasoner (Medium)	Play	92.26	77.5	450.2
SkillReasoner (High)	Play	94.94	74.5	439.3
GCIQL	Play	1.33	0.0	720.0
HIQL	Play	1.73	0.0	720.0
$\pi_{0.5}$	Play	1.92	0.0	720.0
ACT	Play	4.43	0.0	720.0
DP	Play	2.17	0.0	720.0
$\pi_{0.5}$	Expert	3.70	0.0	720.0
ACT	Expert	6.36	0.0	720.0
DP	Expert	15.73	0.0	720.0
`multi-room` **(full)**				
SkillReasoner (Low)	Play	72.02	26.0	661.7
SkillReasoner (Medium)	Play	81.90	63.0	529.9
SkillReasoner (High)	Play	82.12	53.0	556.5
GCIQL	Play	0.82	0.0	720.0
HIQL	Play	1.11	0.0	720.0
$\pi_{0.5}$	Play	1.15	0.0	720.0
ACT	Play	0.87	0.0	720.0
DP	Play	0.73	0.0	720.0
$\pi_{0.5}$	Expert	6.51	0.0	720.0
ACT	Expert	3.81	0.0	720.0
DP	Expert	8.13	0.0	720.0
`winding-maze` **(full)**				
SkillReasoner (Low)	Play	71.19	11.5	703.4
SkillReasoner (Medium)	Play	76.87	24.0	675.6
SkillReasoner (High)	Play	87.44	34.0	653.3
GCIQL	Play	0.30	0.0	720.0
HIQL	Play	1.40	0.0	720.0
$\pi_{0.5}$	Play	0.89	0.0	720.0
ACT	Play	0.56	0.0	720.0
DP	Play	0.60	0.0	720.0
$\pi_{0.5}$	Expert	24.83	0.0	720.0
ACT	Expert	28.07	0.0	720.0
DP	Expert	47.58	0.0	720.0

Table 7 reports the additional passage readouts from the same full `winding-maze` rollouts. Each measures the maximum route progress toward a passage exit, reaching full credit only when the entire T-block clears that passage. The policy receives the final goal beyond all three passages throughout; these readouts do not measure final-pose success.

Table 7: **Progress through successive passages.** First- and second-passage scores (%) from the full `winding-maze` trials in Table 6.

Method	Data	First passage	Second passage
SkillReasoner (Low)	Play	99.51	99.16
SkillReasoner (Medium)	Play	100.00	99.02
SkillReasoner (High)	Play	100.00	99.78
GCIQL	Play	3.81	1.21
HIQL	Play	13.19	4.92
$\pi_{0.5}$	Play	9.71	3.35
ACT	Play	5.78	2.05
DP	Play	6.45	2.32
$\pi_{0.5}$	Expert	91.64	77.73
ACT	Expert	84.93	76.67
DP	Expert	94.06	87.38