Proceedings of the Nineteenth International Symposium on Combinatorial Search (SoCS 2026)

MOSAIC: Skill-Centric Manipulation Planning with Physics Simulation (Extended Abstract)

Itamar Mishani, Yorai Shaoul, Maxim Likhachev

Robotics Institute, School of Computer Science
Carnegie Mellon University, United States
{imishani, yoraish, maxim}@cs.cmu.edu

Introduction

Recent progress in robotics has enabled learning and designing complex manipulation skills, yet deploying them requires planning—deciding which skill to use, where and how to apply it to generalize beyond training, decomposing goals into achievable subtasks, and sequencing skills into coherent solutions—a robot tidying a cluttered table must autonomously reorient plates for stable grasps, avoid obstacles, and coordinate diverse motions, demanding reasoning over long skill sequences and identifying when each imperfect skill is likely to succeed. Existing approaches follow two extremes: policy learning excels at short-horizon tasks but fails at long-horizon reasoning, while Task and Motion Planning (TAMP) offers hierarchical reasoning but is constrained by brittle symbolic representations that are laborious to construct. Moreover, most planning methods follow a sequential search paradigm—from the start state or backward from the goal—which degrades when solutions depend on non-obvious intermediate steps that are required to achieve the task but offer no immediate indication of progress toward it.

We introduce , a framework for long-horizon manipulation planning that positions skills as active participants in the search and uses a physics simulator as a world model to evaluate their feasibility. Rather than following directed search, MOSAIC explores multiple directions simultaneously, focusing its search in regions where skills are most likely to succeed. It employs **Generator Skills** that propose physically grounded parameterized skill-trajectories, identifying "islands of competence," and **Connector Skills** that link these islands by solving boundary value problems, together forming a graph from which solution paths discovered.

MOSAIC

MOSAIC constructs a directed multigraph—the *mosaic graph*—where nodes are generator skill-trajectories and edges are connector transitions (Fig. 1). An *oracle* module orchestrates graph construction, balancing exploration (new generator nodes) and exploitation (connecting nodes toward a solution path).

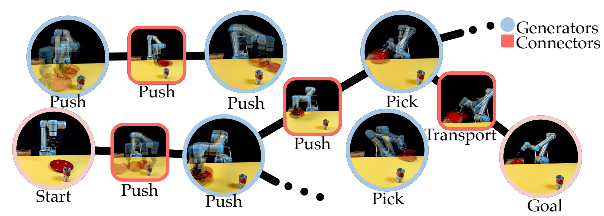


Figure 1: MOSAIC builds a directed multigraph over skill trajectories. Nodes (circles) are generator trajectories—"islands of competence." Edges (squares) are connector skills solving boundary value problems between nodes. The oracle orchestrates construction.

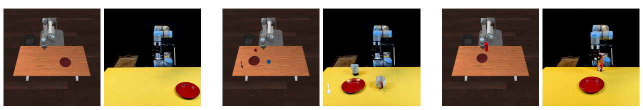
Notation. Let $\mathcal{X} = \mathcal{Q}_R \times \mathcal{Q}_{\mathcal{O}_1} \times \cdots \times \mathcal{Q}_{\mathcal{O}_k}$ be the world state space (robot configuration $\times$ object poses); let $\Xi = \{\xi \mid \xi : \mathcal{X} \to \{0,1\}\}$ be the set of *boundary conditions* (a specific state x' induces $\mathbf{1}_{\{x'\}} \in \Xi$ as a special case) and $\xi_g \in \Xi$ the goal condition. A *skill* σ maps parameters $\theta \in \Theta_\sigma$ to a trajectory $\tau : [0,1] \to \mathcal{X}$, validated by physics-simulation rollout. Generators $\mathcal{G} : \Theta_{\mathcal{G}} \to \mathcal{T}$ produce trajectories without boundary constraints; Connectors $\mathcal{C} : \Xi \times \Xi \times \Theta_{\mathcal{C}} \to \mathcal{T}$ solve boundary value problems given start and goal conditions. A valid solution is a chain $\{\tau_1, \ldots, \tau_N\}$ with $\tau_1(0) = x_s$, $\xi_g(\tau_N(1)) = 1$, and $\tau_i(1) = \tau_{i+1}(0)\ \forall i$.

Algorithm 1 gives the pseudocode: generator nodes seed the graph (lines 3–7); the main loop (lines 8–19) adds oracle-selected connector edges and generator nodes, each estimated via physics-simulation; the result is the minimum-cost path from x_s to any goal-satisfying state.

Oracle. The oracle balances exploration and exploitation based on the node-to-edge ratio in the graph: a high ratio favors connectors; a low ratio favors generators. Each skill is scored by $U(\sigma_i) = \alpha s_{\sigma_i} + (1-\alpha)\sqrt{\ln \frac{\sum_j (t_{\sigma_j}+1)}{t_{\sigma_i}+1}} + n$ (s_σ: success rate; t_σ: invocation count; $n \sim \mathcal{N}(0,1)$), combining past performance with an upper-confidence exploration bonus. Connector boundary conditions are chosen by random pairing (exploration) or nearest-neighbor (exploitation); repeatedly-failed pairs are penalized to avoid wasted effort. The oracle requires no task-specific tuning, making MOSAIC directly applicable to new skill libraries.

Algorithm 1 MOSAIC

Require: start x_s, goal ξ_g, skill library Σ, oracle O
$\mathcal{M} \leftarrow$ empty graph; $\mathcal{M}$.AddNode(x_s)
for each generator $\mathcal{G} \in \Sigma$ **do**
  $\tau \leftarrow \mathcal{G}(O.\text{Sample}(\mathcal{G}))$ {sim-validated}
  if $\tau \neq \emptyset$ **then**
    $\mathcal{M}$.AddNode($\mathcal{G}, \tau$)
  end if
end for
while $\neg$ HasPath($\mathcal{M}$, x_s, ξ_g) **do**
  σ, $\theta \leftarrow O$.ChooseSkill(Σ, $\mathcal{M}$)
  if σ is Connector **then**
    $\xi_0, \xi_1 \leftarrow O$.ChooseBoundaries($\mathcal{M}$)
    $\tau \leftarrow \sigma(\xi_0, \xi_1, \theta)$ {sim-validated}
    if $\tau \neq \emptyset$ **then**
      $\mathcal{M}$.AddEdge($\sigma, \xi_0, \xi_1, \tau$)
    end if
  else
    $\tau \leftarrow \sigma(\theta)$ {Generator}
    if $\tau \neq \emptyset$ **then**
      $\mathcal{M}$.AddNode(σ, τ)
    end if
  end if
end while
return ShortestPath($\mathcal{M}$, x_s, $\{x \mid \xi_g(x)=1\}$)



(a) Transport (b) Transport in Clutter (c) Transport Among Movable Objects

Figure 2: Experimental scenarios (simulation left, real-world right). In all scenarios, the robot must place the plate into the bin.

Experiments

We evaluate MOSAIC on three manipulation scenarios (Fig. 2) using the same four-skill library: **Push** ($\mathcal{G}$+$\mathcal{C}$), a learned diffusion push policy (Chi et al. 2024) ; **Pick** ($\mathcal{G}$), a geometry-based grasp skill; **Transport** ($\mathcal{C}$), a motion-planning transport skill; and **Rearrange** ($\mathcal{C}$, scenario (c) only), a multi-object repositioning skill. All skills are validated via rollout in the SAPIEN physics engine. *Transport*: the plate sits deep on the table and cannot be grasped directly. The planner must discover it needs to first push the plate to the table's edge before grasping it. *Transport in Clutter*: static obstacles surround the plate. The robot must thread between them without collisions. *Transport Among Movable Objects*: a second object may block plate access. The planner must discover whether to clear it or work around it—expanding the search space to multi-object interactions.

We compare against four baselines: **Skills as Options** (Konidaris and Barto 2009) (sequential forward chaining), **CEM** (Rubinstein and Kroese 2004) (receding-horizon parameter optimization), **MOSAIC-Roadmap** (PRM-style (Kavraki et al. 1996) batch roadmap construction without oracle guidance), and **Incremental MOSAIC-Roadmap** (online extension of the same). Results are shown in Table 1. MOSAIC achieves 75–100% success across all scenarios while consistently outperforming baselines; sequential baselines (Skills as Options, CEM) degrade sharply on complex tasks due to their inability to reason about non-obvious mandatory intermediate steps. The roadmap ablations reveal that flexible non-sequential exploration helps, but oracle-guided adaptive construction is essential for consistent performance. Hardware demonstrations on a real UR10e robot with RGB-D perception can be found at skill-mosaic.github.io.

	MOSAIC	MOSAIC-Roadmap	Incremental MOSAIC-Roadmap	Skills as Options	CEM
Scenario (a): Plate Transport					
Success Rate [%]	100	100	94	72	35
Average Planning Time [sec]	97.9	602.4	360.4	535.2	919.3
Scenario (b): Plate Transport in Clutter					
Success Rate [%]	85	65	80	50	0
Average Planning Time [sec]	286.7	737.2	459.7	486.4	–
Scenario (c): Plate Transport Among Movable Obstacles					
Success Rate [%]	75	65	55	20	0
Average Planning Time [sec]	423.2	788.4	685.8	952.0	–

Table 1. Success rates and average planning times across all three scenarios.

Conclusion

MOSAIC introduces a skill-centric planning paradigm in which graph construction and solution-finding are unified and non-sequential: it iteratively expands a graph anchored in regions where skills are likely to succeed until a feasible path from start to goal emerges.

Combined with physics simulation for feasibility estimation and a statistics-driven oracle for adaptive guidance, MOSAIC robustly solves long-horizon manipulation tasks that challenge existing sequential baselines. Promising future directions include integrating foundation models as high-level oracles, extending to partial observability, and exploiting massively parallel physics simulation to accelerate skill evaluation.

References

Chi, C.; Xu, Z.; Feng, S.; Cousineau, E.; Du, Y.; Burchfiel, B.; Tedrake, R.; and Song, S. 2024. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. *The International Journal of Robotics Research*.

Kavraki, L.; Svestka, P.; Latombe, J.-C.; and Overmars, M. 1996. Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. *IEEE Transactions on Robotics and Automation*, 566–580.

Konidaris, G.; and Barto, A. 2009. Skill discovery in continuous reinforcement learning domains using skill chaining. *Advances in Neural Information Processing Systems*, 22.

Rubinstein, R. Y.; and Kroese, D. P. 2004. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*. Springer.