Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

KinDER: A Physical Reasoning Benchmark for Robot Learning and Planning

Yixuan Huang[1*], Bowen Li[2*], Vaibhav Saxena[3*], Yichao Liang[4], Utkarsh A. Mishra[3], Liang Ji[1], Lihan Zha[1], Jimmy Wu[5], Nishanth Kumar[6], Sebastian Scherer[2], Danfei Xu[3,5], Tom Silver[1]

[1]Princeton University, [2]Carnegie Mellon University, [3]Georgia Tech, [4]University of Cambridge, [5]NVIDIA, [6]MIT

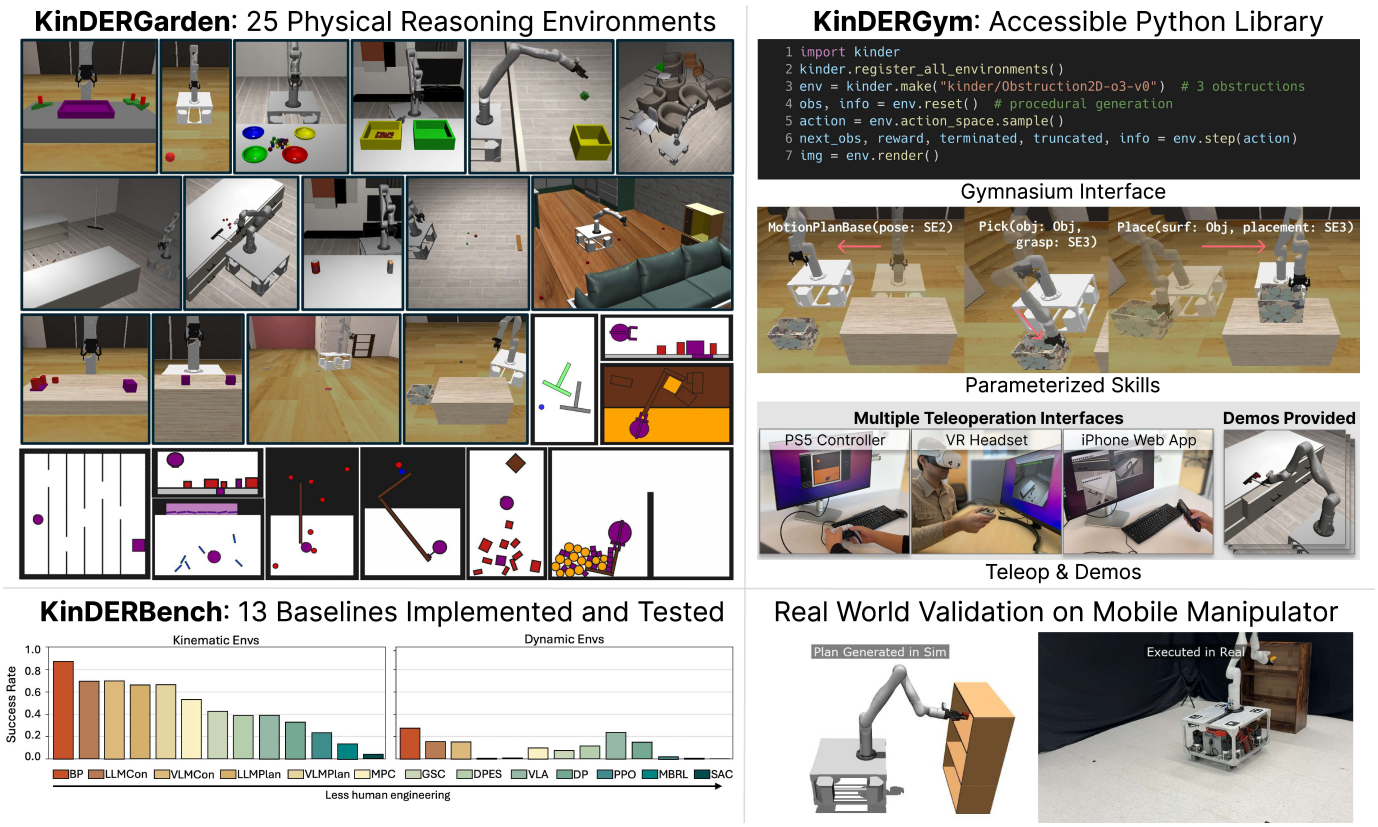

Fig. 1: **KinDER Overview.** We present **KinDER**, a physical reasoning benchmark for robot learning and planning with three main contributions: **KinDERGarden** (top left), a collection of 25 physical reasoning environments; **KinDERGym** (top right), a Python library with a Gymnasium interface, parameterized skills, multiple teleoperation interfaces, and demonstrations; and **KinDERBench** (bottom left), a suite of baselines and evaluations. We also show real-world validity on a mobile manipulator (bottom right).

***Abstract*—Robotic systems that interact with the physical world must reason about kinematic and dynamic constraints imposed by their own embodiment, their environment, and the task at hand. We introduce KinDER, a benchmark for Kinematic and Dynamic Embodied Reasoning that targets physical reasoning challenges arising in robot learning and planning. KinDER comprises 25 procedurally generated environments, a Gymnasium-compatible Python library with parameterized skills and demonstrations, and a standardized evaluation suite with 13 implemented baselines spanning task and motion planning, imitation learning, reinforcement learning, and foundation-model-based approaches. The environments are designed to isolate five core physical reasoning challenges: basic spatial relations, nonprehensile multi-object manipulation, tool use, combinatorial geometric constraints, and dynamic constraints, disentangled from perception, language understanding, and application-specific complexity. Empirical evaluation shows that existing methods struggle to solve many of the environments, indicating substantial gaps in current approaches to physical reasoning. We additionally include real-to-sim-to-real experiments on a mobile manipulator to assess the correspondence between simulation and real-world physical interaction. KinDER is fully open-sourced and intended to enable systematic comparison across diverse paradigms for advancing physical reasoning in robotics. Website and code: https://prpl-group.com/kinder-site/**

*Equal contribution. Correspondence to yh1542@princeton.edu

I. Introduction

A central challenge in robotics that arises from embodied interaction with the real world is the need for *physical reasoning* [1–8]. Broadly competent robots must be able to reason about the kinematic and dynamic limits dictated by their own morphology, the laws of physics imposed by their environment, and the task requirements specified by their users. These often-entangled constraints can turn semantically

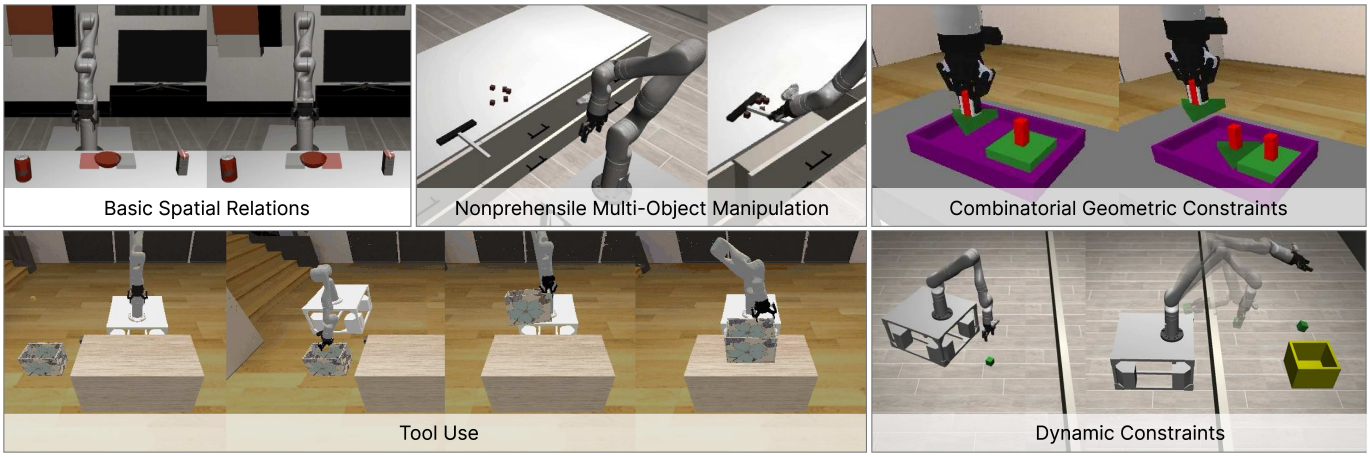

Fig. 2: **Core Challenges for Physical Reasoning in KinDER.** From top left: arranging objects to be in goal-specified locations relative to a bowl requires understanding *basic spatial relations*. Sweeping many small objects into a drawer benefits from *nonprehensile multi-object manipulation*. Packing varying numbers of objects into a confined region requires satisfying *combinatorial geometric constraints*. Transporting objects with a box benefits from *tool use*. Tossing objects over a barrier requires satisfying *dynamic constraints*.

simple tasks into challenging puzzles [3, 9, 10]. To store a book in a shelf, is it enough to move to the book, grasp it, move to the shelf, and place? That depends: is there a clear path to the book, or do obstacles need to be moved? Does the book need to be set down and re-grasped before a placement is feasible? Is there space in the shelf, or should the robot use its arm to gently push other books aside? The robot must not only answer these questions, but pose them in the first place—*reasoning about what to reason about* [11–13] given only the available sensory observations.

Despite the importance of physical reasoning to robotics, there is little consensus on the state of the art. Measuring physical reasoning is hard: no single task is sufficient (why not just memorize the solution?) and even procedurally-generated variations [14–16] of a task cannot capture the challenge of physical reasoning in its full generality. Existing benchmarks (Section III) cover more general challenges for robot learning and planning—broad task diversity, long-horizon decision making, language grounding—or focus on full-fledged application-focused domains such as home assistance. As a result, it remains difficult to perform targeted evaluation of physical reasoning itself, disentangled from perception, language understanding, or domain-specific considerations.

Another reason for the lack of consensus is that physical reasoning has been studied from very different perspectives in separate subfields of robotics. Classical approaches such as task and motion planning (TAMP) [3, 10, 33–35] use explicit models and optimization techniques to formulate and solve generalized constraint satisfaction problems. Model-free approaches such as reinforcement learning (RL) [36–38] and imitation learning (IL) [39–41] use data to compile away the need for explicit reasoning. Foundation model (FM) based approaches such as LLM [5, 42], VLM [6, 43], or VLA [7, 44, 45] planning combine explicit reasoning in natural language with implicit understanding from pretraining. There is also broad interest in combining the complementary strengths of these approaches, but without clarity on the state of the art, it is difficult to make progress.

To address these challenges, we propose (**KinDER**): a benchmark for **Kin**ematic and **D**ynamic **E**mbodied **R**easoning. KinDER has three main contributions (Figure 1):

1) **KinDERGarden**: A collection of 25 simulated environments, each with infinite procedurally-generated variations, to capture different facets of physical reasoning.
2) **KinDERGym**: A Python package that includes a Gymnasium-compatible environment API, a collection of parameterized skills and concepts, multiple teleoperation interfaces, and demonstration datasets.
3) **KinDERBench**: A standardized benchmark for physical reasoning approaches, with 13 pre-implemented baselines from the literature on TAMP, RL, IL, and FM.

All contributions are open-source and tested on multiple standard operating systems. To show that the simulated environments map onto real physical reasoning challenges, we additionally report real-to-sim-to-real results. Taken together, KinDER represents a significant step toward clarifying and advancing the state-of-the-art in robot physical reasoning.

II. KinDER Core Challenges

We begin by presenting the core physical reasoning challenges that are prioritized in KinDER. To select these challenges, we started by reviewing (1) existing work in robot planning and learning where individual physical reasoning problems are considered with one-off environments; and (2) existing benchmarks in related areas (Section III). We then identified themes in (1) that are not well-represented in (2). In other words, we chose challenges at the frontier of active research, but where the current state-of-the-art remains unclear.

The five KinDER core challenges are illustrated by example in Figure 2. In Section III, we discuss coverage of these challenges in existing benchmarks. In Section IV, we detail how environments in KinDERGarden capture the challenges. The challenges are as follows:

Benchmark	Focus On Physical Reasoning	2D/3D	Basic Spatial Relations	Nonprehensile Multi-Object Manipulation	Tool Use	Combinatorial Geometric Constraints	Dynamic Constraints
BEHAVIOR-1k [17]		3D	✓		✓	✓	✓
CALVIN [18]		3D	✓				
DittoGym [19]	✓	2D		✓			
DM Control [20]	✓	3D					✓
Embodied Interface [21]		3D	✓		✓		
EmbodiedBench [22]		3D	✓		✓		
FurnitureBench [23]		3D			✓	✓	
I-PHYRE [24]	✓	2D			✓		
Kinetix [25]		2D			✓		✓
Lagriffoul et al. [26]	✓	Both			✓		
LIBERO [27]	✓	3D	✓			✓	
ManiSkill-HAB [28]		3D	✓			✓	
OGBench [29]		Both					✓
RoboCasa [30]		3D				✓	✓
Virtual Tools [9]	✓	2D			✓		✓
VLABench [31]		3D	✓		✓	✓	
VLMgineer [32]	✓	3D		✓	✓		✓
KinDER (Ours)	✓	Both	✓	✓	✓	✓	✓

TABLE I: **Related Benchmarks.** KinDER is a benchmark for robot physical reasoning with both 2D and 3D environments and with an emphasis on five core challenges (Section II). KinDER fills a gap in the literature; see Section III for discussion.

1) **Basic Spatial Relations:** To set a dinner table [46], load a dishwasher [47], or follow instructions with locative prepositions [48], robots must understand spatial relations between objects. They must have both a passive understanding (is the fork on the left of the plate?) and an active understanding (how can I put it there?).
2) **Nonprehensile Multi-Object Manipulation:** Generalized manipulation requires more than pick and place—robots should be able to push [49], pull, sweep [50], scoop, stir, and slap [51] multiple objects at the same time. They should leverage, rather than strictly avoid, whole-arm [52] and whole-body [53] contact.
3) **Tool Use:** Robots should use objects to manipulate other objects—hammers [54], wrenches [55], hooks [56], sticks [57], trays [58], bins [59], and step-stools [60]. They should understand not only common tool affordances, but also abstract mechanisms so that they can improvise [61], e.g., use a rock to pitch a tent [9].
4) **Combinatorial Geometric Constraints:** When object-object and robot-object collisions need to be avoided, e.g., while packing [62], retrieving [63], or navigating around [64] objects in tight and cluttered spaces, robots must understand and work within implicit geometric constraints. These constraints are combinatorial [3]: when the number of objects grows, the number of constraints (e.g., pairwise collisions) grows polynomially.
5) **Dynamic Constraints:** To carry a full cup of coffee [65], balance a delicate tray, scoop and pour without spilling [66], dribble and toss a basketball [67], or juggle [68], robots must use control to stabilize dynamical systems. They should understand and obey dynamic constraints, e.g., safety limits on velocity magnitudes or requirements implicit in a task (don't spill).

This list is by no means an exhaustive account of all the challenges associated with robot physical reasoning. Nonetheless, progress on these challenges would represent a significant step forward for the field. It is also worth noting that more general decision-making challenges are pervasive in KinDER—long task horizons, sparse feedback (goal-based rewards), broad task distributions, and time pressure during planning and execution. We omit these from the core list above to keep the focus on physical reasoning.

III. Related Work

We next discuss existing benchmarks and their relationship to KinDER. The main novelty of KinDER is its coverage of the core physical reasoning challenges introduced in Section II. These challenges are the *focus* in KinDER, as opposed to application-driven benchmarks (e.g., home assistance) where physical reasoning is entangled with other factors. KinDER also includes both 2D and 3D environments that permit study of physical reasoning at multiple levels of abstraction. See Table I for a summary of related work.

a) Benchmarks for Robot Learning and Planning: There has been a significant amount of work on benchmarking robot learning methods [17, 18, 20–23, 27–32]. Some benchmarks are geared toward imitation learning [69] *or* reinforcement learning [20, 25, 29] *or* foundation-model-based methods [21, 22, 31]; others are explicitly designed to compare different families of techniques. Table-top manipulation is a common setting [18, 23, 27], but mobile [17, 28, 30] and bimanual [70] manipulation are also considered. The central technical challenges in these benchmarks include long time horizons, sparse rewards, natural language grounding, and broad task diversity (especially in terms of scene and object variation). For KinDER, we especially take inspiration from LIBERO [27] and MimicLabs [69].

There is far less work on benchmarks for classical robot planning (e.g., task and motion planning). There are also separate benchmarks for motion planning [71–73] and task planning [74–76]. In particular, the International Planning Competition [74–76] has been a longstanding catalyst for task planning research. To the best of our knowledge, the only benchmark for combined TAMP is the one proposed by Lagriffoul et al. [26], which is not actively used. KinDER facilitates direct comparisons between robot planning and robot learning methods, and their combinations: KinDERGym provides parameterized skills and concepts, and KinDER-Bench reports results for both planning and learning methods.

b) Application-Driven Benchmarks: KinDER isolates fundamental challenges of physical reasoning so that researchers can get a clear signal as they work on these challenges. In this sense, KinDER is complementary to benchmarks that are explicitly driven by applications. Home assistance applications are especially well-covered by benchmarks such as ALFRED [77], AI2-THOR [78] and ManipulaTHOR [79], BEHAVIOR-1k [17], Habitat [80], ManiSkill-HAB [28], and RoboCasa [30]. Other notable and recent application-focused benchmarks include FurnitureBench [23] for furniture assembly, CleanUpBench [81] for sweeping and grasping, and CookBench [82] for cooking. The need for physical reasoning naturally arises in these benchmarks, among many other challenges for robot perception, planning, and learning. KinDER is designed to evaluate and advance robot physical reasoning specifically.

c) Physical Reasoning Benchmarks: KinDER takes inspiration from benchmarks outside of robotics that focus on physical reasoning such as the Virtual Tools Game [9] and PHYRE [83]. See Melnik et al. [84] for a survey. In contrast to many of these works, our intention is to advance robotics, rather than to better understand human physical reasoning. Nonetheless, drawing connections between KinDER and human-like physical reasoning approaches represents an opportunity for future work [85]. From the perspective of this literature, important aspects of KinDER include: continuous spaces, multi-step (long-horizon) decision-making, procedural generation, and kinematic and dynamic constraints.

IV. KinDERGarden: Environments

Our first contribution is KinDERGarden, a collection of 25 environments for robot physical reasoning, grouped into four categories: Kinematic2D, Dynamic2D, Kinematic3D, and Dynamic3D. We first discuss what is common among all environments and then describe each category. See Appendix A for details and Figure 3 for KinDER core challenge coverage.

a) General Environment Structure: KinDERGarden environments inherit from the general Gymnasium [86] API, which includes an observation space, action space, initial state distribution $\mathrm{reset}()$, and a $\mathrm{step}()$ function that takes an action as input and produces a next observation, reward, and termination indicator. Rewards are sparse: -1 is given at every step until successful termination, which occurs only when a goal is achieved. All environments have an infinite task distribution that is implemented with procedural generation inside the $\mathrm{reset}()$ function; see Figure 4 for an example.

The main design decision that distinguishes KinDER from the general Gymnasium API is that all environments use *object-centric states*. An object-centric state is a mapping from object names (e.g., robot, hook) to real-valued feature vectors. The dimensionality of each vector is determined by object *type*. For example, a robot with type $\mathrm{MobileManipulator}$ has features for the robot's base position and velocity in $\mathrm{SE}(2)$, arm configuration and velocity in $\mathbb{R}^7$, and gripper joint value in $[0, 1]$. A hook with type $\mathrm{Movable}$ has features for pose and velocity in $\mathrm{SE}(3)$ and bounding box dimensions in $\mathbb{R}^3$, among others. Another $\mathrm{Movable}$ object (e.g., a plate) would have the same feature space. This design makes it easy to vary the number of objects, which can be useful for evaluating generalization and test-time scaling (Section VI).

Baselines in KinDER can use object-centric states directly, but to facilitate experiments with standard learning-based approaches, we provide two other options. The first option is to use RGB image observations. The second is to commit to a *variant* of a KinDERGarden environment where the objects are constant. For example, in $\mathrm{Shelf3D}$, the number of books can vary in general, but in the $\mathrm{b5}$ variant, there are always 5 books. For constant-object variants, KinDERGarden flattens the object-centric state into a fixed-dimensionality vector. These environments are then compatible with standard reinforcement learning and imitation learning approaches.

b) Kinematic2D Environments: The Kinematic2D category includes six environments that are especially useful for studying tool use and combinatorial geometric constraints at a high level of abstraction. This category is *kinematic* in the sense that environment transitions are entirely determined by object poses and robot configurations (velocities and accelerations are not modeled); and *2D* in that it is implemented with 2D shapes. All environments have a robot with a circular base that moves in $\mathrm{SE}(2)$, an extendable 1D arm, and a rectangular vacuum on its end effector that can be activated or deactivated. When the vacuum is activated, all objects in its immediate vicinity become rigidly attached to the robot. Actions are constrained to make small changes to the robot's configuration. When an action is received, a tentative next state is computed. If that next state includes any collisions, the state is reverted. These environments are implemented in pure Python; no physics backend is used.

c) Dynamic2D Environments: The Dynamic2D category includes four environments that are especially useful for studying nonprehensile multi-object manipulation and tool use at a high level of abstraction. Unlike Kinematic2D, velocities and accelerations are modeled in this category. We use the Pymunk physics backend for dynamics [87]. Similar to Kinematic2D, these environments feature a robot with a circular base and an extendable 1D arm. For the benefit of studying contact-rich dynamics, we use a two-fingered gripper on the end effector.

Kinematic2D and Dynamic2D environments require qualitatively different forms of physical reasoning (Figure 5). For example, consider the contrast between $\mathrm{Obstruction2D}$ (kine-

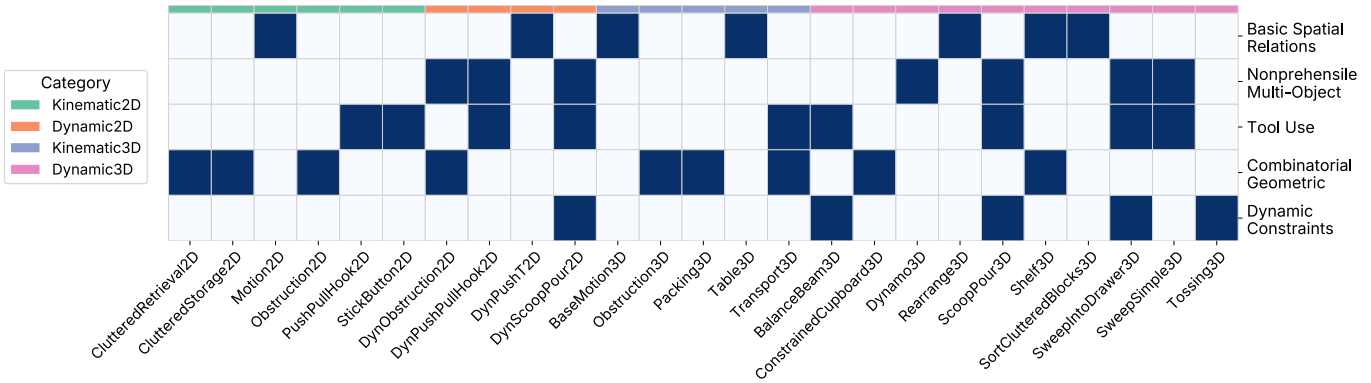

Fig. 3: **KinDERGarden Core Challenges.** Environments in KinDERGarden cover the five core challenges for physical reasoning.

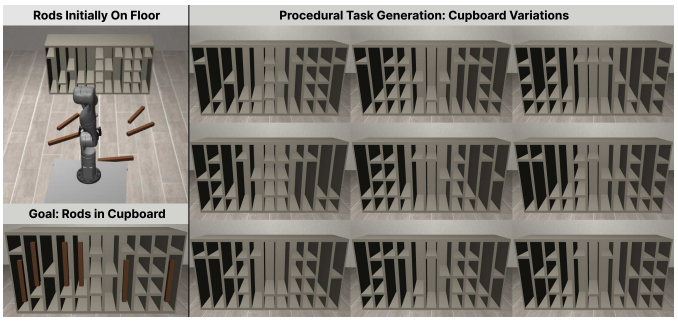

Fig. 4: **Procedural Task Generation Example.** Shelves are randomly arranged within the cupboard in ConstrainedCupboard3D, forcing the robot to reason about the feasibility of rod placements.

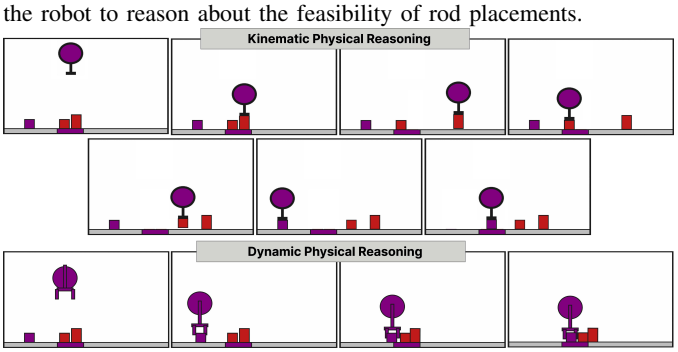

Fig. 5: **2D Kinematic and Dynamic Physical Reasoning Examples.** In Obstruction2D, the robot must pick and place obstacles to make space on a target region. In DynObstruction2D, the robot can push the obstacles out of the way while grasping the target object.

matic) and DynObstruction2D (dynamic). In both environments, the goal is to move a target object onto a target region that may be initially obstructed by one or more obstacles. In the kinematic version, the robot has no choice but to pick and place the obstacles before picking and placing the target object. However, in the dynamic version, shortcuts [51] are possible: if space constraints allow, the robot may be able to push the obstacles out of the way while holding the target.

d) Kinematic3D Environments: The Kinematic3D category includes five environments that are especially useful for studying spatial relations and combinatorial geometric constraints. These environments are kinematic in the same sense as Kinematic2D (no velocities or accelerations). We use object modeling, forward kinematics, and collision-checking methods from PyBullet [88] to implement transitions in this environment. For consistency, all environments feature a TidyBot++ [89] mobile base with a 7DOF Kinova Gen3 arm [90] and a Robotiq 2F-85 [91] gripper. When the gripper is closed, objects between the fingers become rigidly attached to the robot until the gripper is opened. As with Kinematic2D, actions are constrained to make small changes to the robot's configuration; states are reverted when collisions are detected.

e) Dynamic3D Environments: The Dynamic3D category includes 10 environments that collectively cover all five core physical reasoning challenges. Velocities and accelerations are modeled; we use the MuJoCo physics backend for dynamics [92]. For consistency, we use the same TidyBot++ mobile manipulator as in Kinematic3D. Unlike Kinematic3D, grasping is dynamic—objects are never rigidly attached to the robot. Inspired by other MuJoCo-based benchmarks such as LIBERO [27], we use environment configuration files so that all Dynamic3D environments share the same Python code and differ only in their configurations. We also take inspiration from the BDDL specification language introduced in BEHAVIOR [17] in our implementation of procedural task generation, and leverage object and scene assets from RoboCasa [30] and MimicLabs [69] respectively.

V. KinDERGym: Accessible Software

Our second main contribution is KinDERGym, a pip-installable Python package that includes not only an interface to the environments in KinDERGarden, but also (1) parameterized skills and concepts; (2) teleoperation interfaces; and (3) precollected demonstrations. To facilitate ease of use, we developed KinDERGym following strict software engineering standards including continuous integration, linting, type checking, autoformatting, and nearly 400 unit tests. We have tested Python versions 3.10, 3.11, 3.12, Ubuntu 20.04, 22.04, and 24.04, and macOS 12-15, and Windows 10.

a) Parameterized Skills and Concepts: KinDERGym provides utilities for defining parameterized skills and concepts that can be used for hierarchical planning and learning. Skills are implemented as options [93] with associated PDDL operators [94] and samplers [35]. The options have both object parameters (the same as the PDDL operator) and additional parameters of any type (proposed by the sampler). For example, a $\mathrm{Pick}(\mathrm{object}, \theta)$ skill can be used to pick different objects

with different relative grasps $\theta \in \mathrm{SE}(3)$. For generality, we allow option policies to maintain internal state. A common pattern is to generate and follow a motion plan.

Concepts are implemented as relational predicates with classifiers that ground in object-centric states [95, 96]. For example, $\mathrm{On}(\mathrm{object}, \mathrm{surface})$ is a predicate with a classifier that evaluates to True in states where the object is above and in contact with the $\mathrm{surface}$. These predicates are used in the preconditions and effects of the skill operators. Together with the object-centric states, concepts can also be understood as defining a two-level scene graph [97].

In our experiments (Section VI), we use KinDERGym skills and concepts for the bilevel planning, LLM planning, and VLM planning baselines. However, the nature of physical reasoning is such that hierarchical task decompositions are not always readily apparent or easy to engineer. Designing or learning such skills remains an important direction for future work on physical reasoning that KinDER can support.

b) Teleoperation Interfaces and Demonstrations: KinDERGym includes multiple teleoperation interfaces that can be used to collect human demonstrations. Kinematic2D and Dynamic2D environments can be controlled through a mouse-and-keyboard interface, or through a PS5 video game controller. The mouse-and-keyboard interface includes joystick-like buttons that can be clicked and dragged to move the robot in $\mathrm{SE}(2)$. Keyboard commands extend and retract the arm, activate and deactivate the vacuum (for Kinematic2D), and open and close the gripper (for Dynamic2D). The PS5 controller similarly uses the joysticks to move the robot and buttons for the arm, vacuum, and gripper.

Kinematic3D and Dynamic3D environments can be controlled through an iPhone web app, or through a Meta Quest 3S virtual reality headset and controller. The iPhone web app is based on the TidyBot++ interface [89], which uses the iPhone's gyroscope and accelerometer to capture spatial inputs. The teleoperator can toggle between base and arm control. For arm control, inputs are mapped to task (end effector) space and inverse kinematics is used to derive environment actions. The Meta Quest 3S interface uses the right-hand controller for task-space inputs and the left-hand controller for base movements. We additionally allow the teleoperator to select one or more camera angles, which can also be defined relative to the robot.

For environments with parameterized skills and concepts implemented, we can also use planners to derive demonstrations at scale. As part of KinDERGym, we use a combination of teleoperation and planning to provide ≥ 100 precollected demonstrations for 10 environments (Appendix B). We encourage KinDER users to collect and open-source additional demonstrations using the teleoperation interfaces provided.

VI. KinDERBench: Baselines and Metrics

Our third contribution is KinDERBench, a standardized multi-metric benchmark for robot physical reasoning. We report results and release implementations for 13 baselines in 8 environments. In this section, we briefly describe the environments, baselines, and metrics, analyze the results, and present insights from additional experiments.

a) Environments: We select two representative environments with varying levels of difficulty from each of the four KinDERGarden categories. See Appendix A for detailed descriptions.

1) $\mathrm{Motion2D}$: The simplest Kinematic2D environment. The robot must move to reach a goal region. In this variant ($\mathrm{p0}$), there are no obstacles.
2) $\mathrm{StickButton2D}$: A Kinematic2D environment where a robot must press a button. The button is sometimes out of reach, requiring the robot to use a stick as a tool to press it. In this variant ($\mathrm{b1}$), there is one button.
3) $\mathrm{DynObstruction2D}$: The Dynamic2D environment in Figure 5. In this variant ($\mathrm{o1}$), there is one obstacle.
4) $\mathrm{DynPushPullHook2D}$: A Dynamic2D environment that requires using a hook to pull a target object surrounded by obstacles. In this variant ($\mathrm{o5}$), there are five obstacles.
5) $\mathrm{BaseMotion3D}$: The simplest Kinematic3D environment. The robot must move its base to reach a goal region. In this variant ($\mathrm{o0}$), there are no obstacles.
6) $\mathrm{Transport3D}$: A Kinematic3D environment where a box and one or more objects must be moved from the floor to a table. The box may be used as a container, but this is not required (and not always optimal). In this variant ($\mathrm{o2}$), there are two objects in addition to the box.
7) $\mathrm{Shelf3D}$: A Dynamic3D environment where objects must be packed into a space-constrained shelf. In this variant ($\mathrm{o1}$), one object must be packed.
8) $\mathrm{SweepIntoDrawer3D}$: A Dynamic3D environment where small objects on a countertop must be moved to an initially closed drawer, optionally using a sweeping tool. In this variant ($\mathrm{o5}$), there are 5 objects.

b) Baselines: We evaluate 13 representative planning and learning baselines. See Appendix C for details.

1) **Bilevel Planning (BP)** [34, 95, 96]: A search-then-sample TAMP planner; uses skills and concepts.
2) **LLM Planning (LLMPlan)** [98, 99]: An LLM (GPT-5.2 [100]) planner; uses object-centric states and skills.
3) **VLM Planning (VLMPlan)** [43]: A VLM (GPT-5.2 [100]) planner; uses *RGB images*, states, and skills.
4) **LLM In-context (LLMCon)** [98, 99]: An LLM (GPT-5.2 [100]) planner that is prompted with in-context examples and uses object-centric states and skills.
5) **VLM In-context (VLMCon)** [43]: A VLM (GPT-5.2 [100]) planner that is prompted with in-context examples; uses *RGB images* along with states and skills.
6) **Model Predictive Control (MPC)** [101]: We perform predictive sampling trajectory optimization using MPC with the ground-truth simulation transition functions and (sparse) reward function.
7) **Model-based Reinforcement Learning (MBRL)** [102]: We use the demonstrations collected to train a neural (state-based) transition model and use MPC to select actions with the same sparse reward function.

8) **Generative Diffusion Planning**: We train and evaluate Generative Skill Chaining (GSC) [103] with our demonstration data annotated with skill labels.
9) **Proximal Policy Optimization (PPO)** [36]: A standard *on-policy* deep reinforcement learning method.
10) **Soft Actor-Critic (SAC)** [37]: A standard *off-policy* deep reinforcement learning method.
11) **Diffusion Policy (DP)** [41]: Imitation learning over RGB images, trained with 100 demos per environment.
12) **DP + Environment States (DPES)** [41]: Same as DP, but with states also provided as input.
13) **Finetuned VLA** [104]: A pretrained $\pi_{0.5}$ VLA finetuned on the same demos as DP.

c) Evaluation Metrics: KinDERBench includes multiple metrics that capture different dimensions of efficiency and effectiveness in physical reasoning. These include:

1) **Success Rate (SR)**: A measure of effectiveness.
2) **Cumulative Rewards (Rwd)**: A measure of efficiency. Recall rewards are -1 until success. This metric is considered only for successful episodes.
3) **Inference Time (Inf-Time)**: Another measure of efficiency. We report per-episode wall-clock time (sec).

Another metric that is equally important but difficult to capture quantitatively is **engineering cost**: the amount of engineering needed to run a method in a new environment. For example, BP has a high engineering cost because it requires skills and concepts; RL has a much lower cost.

d) Benchmark Results: We present our main benchmark results in Table II. All baselines are evaluated over 5 random seeds with 50 evaluation episodes per seed. We report means in the main paper and standard deviations in Appendix C. Overall, BP obtains the highest average success rate (0.57), followed by LLMCon (0.43), VLMCon (0.43), LLMPlan (0.34), VLMPlan (0.34), MPC (0.32), VLA (0.32), GSC (0.26), DPES (0.25), DP (0.24), PPO (0.13), MBRL (0.08), SAC (0.02). The general trend is expected: paying higher engineering costs and spending more inference time leads to dividends in success rates.

We now discuss baseline performance in detail, highlighting some of the more surprising results. First, given that both use the same parameterized skills, the gap between BP and LLMPlan/VLMPlan, especially in the more challenging environments, indicates that there remains room to improve the latter approaches. The comparison between LLMPlan/VLMPlan and LLMCon/VLMCon shows that in-context examples are important for the overall performance. Furthermore, the comparable performance between LLMPlan/VLMPlan suggests that the VLM is not able to meaningfully leverage the images that it receives in addition to the object-centric states.

The imitation learning baselines (DP, DPES, VLA) perform well overall, considering that they do not have access to parameterized skills. Interestingly, the VLA is the only baseline that achieves a non-trivial success rate (0.43) on the DynPushPullHook2D task with 5 obstacles. Recall that this environment requires both tool use and nonprehensile multi-object manipulation. This result is surprising because the 2D rendering and physics is quite different from the data used to pretrain the VLA. Another surprising finding is the nontrivial success rate of DP (0.14) and DPES (0.04) on the long-horizon multi-stage SweepIntoDrawer3D, which requires the robot to open the drawer, grasp the sweeper, and then sweep multiple objects into the drawer. We also provide subtask success rates in Table III. We also see that DPES performs comparably to DP, despite its access to object-centric states. This suggests that DP is not able to meaningfully leverage the states, which is an interesting dual to the LLM/VLM case.

We also find that the MPC baseline performs well, given that it only receives the sparse reward functions. We attribute the good performance to the predictive sampling proposed in [101]. The MBRL performs worse than the MPC baseline, though they use the same planner, which demonstrates that the learned transition model is unreliable.

Finally, we find that the RL baselines (PPO and SAC) perform well only in short-horizon tasks, which is expected given the sparse rewards and lack of inductive bias given to these methods. In Appendix B, we report additional results showing that dense rewards can improve performance, but success rates for RL remain low overall.

e) Additional Results: **Out-of-Distribution Generalization:** We next evaluate generalization to unseen scenarios for the imitation learning baselines (DP, VLA) that do not require environment-specific state vectors in the DynObstruction2D environment. After training with 1 obstacle, we test with 0, 2, and 3 obstacles. Results are shown in Table IV. Although there is some performance degradation, both baselines perform surprisingly well in these out-of-distribution tasks. The VLA is particularly robust, perhaps due to pretraining.

Scaling Bilevel Planning: We finally evaluate the efficiency and effectiveness of bilevel planning in the StickButton2D environment as the number of objects increases. In Table V, we see that the success rate and planning time substantially decrease and increase respectively. This highlights an opportunity for future work that uses learning to improve planning for physical reasoning at scale.

VII. Real World Validation

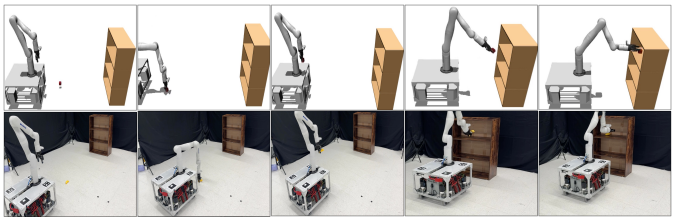

Fig. 6: Real-to-sim-to-real example. We construct a twin simulation from real-world observations using object-centric states, generate motion plans in simulation, and execute them in the real world.

We next demonstrate an example of real-to-sim-to-real with the TidyBot++ as the real robot [89] and the Shelf3D environment in KinDERGarden as the simulator (Figure 6). Our goal is to show that KinDERGarden corresponds to real-world physical reasoning challenges, while also highlighting the potential for future real-to-sim-to-real research using KinDER.

Method	BP	LLMCon	VLMCon	LLMPlan	VLMPlan	MPC	VLA	GSC	DPES	DP	PPO	MBRL	SAC
Motion2D (Kinematic2D)													
SR ↑	**1.00**	**1.00**	**1.00**	0.99	**1.00**	0.92	0.79	**1.00**	0.92	0.88	0.80	0.16	0.00
Rwd ↑	-40.0	-40.0	-40.0	-39.9	-39.9	-92.0	-41.9	-54.9	-45.0	-44.8	-39.6	-186.4	–
Inf-Time ↓	0.07	0.02	2.62	2.72	2.89	29.20	0.62	13.20	0.23	0.28	0.002	72.10	–
StickButton2D (Kinematic2D)													
SR ↑	**0.99**	0.44	0.44	0.25	0.28	0.68	0.53	0.08	0.10	0.10	0.14	0.36	0.00
Rwd ↑	-52.6	-43.3	-46.1	-62.9	-63.2	-90.3	-52.0	-383.2	-48.6	-72.4	-17.0	-149.2	–
Inf-Time ↓	0.85	2.14	2.93	2.49	3.09	117.60	0.97	161.70	0.66	0.67	0.01	280.80	–
DynObstruction2D (Dynamic2D)													
SR ↑	0.08	0.07	0.07	0.02	0.03	0.41	**0.50**	0.32	0.34	0.33	0.08	0.04	0.01
Rwd ↑	-68.62	-82.6	-84.1	-30.2	-5.5	-142.8	-82.7	-408.9	-105.1	-96.4	-29.2	-195.8	-1.0
Inf-Time ↓	15.95	5.39	3.83	3.92	4.19	381.00	1.13	202.50	0.58	0.58	0.01	571.92	0.02
DynPushPullHook2D (Dynamic2D)													
SR ↑	0.01	0.01	0.01	0.00	0.00	0.00	**0.43**	0.00	0.00	0.00	0.00	0.00	0.00
Rwd ↑	-105.3	-117.0	-94.3	–	–	–	-253.8	–	–	–	–	–	–
Inf-Time ↓	73.8	5.58	7.81	–	–	–	3.23	–	–	–	–	–	–
BaseMotion3D (Kinematic3D)													
SR ↑	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.54	0.25	0.66	0.54	0.32	0.00	0.06	0.15
Rwd ↑	-9.9	-9.9	-9.9	-9.9	-9.9	-70.1	-26.5	-93.0	-28.7	-26.8	–	-95.7	-7.8
Inf-Time ↓	0.02	2.54	0.01	2.04	2.16	58.00	0.66	164.80	0.28	0.35	–	125.54	0.003
Transport3D (Kinematic3D)													
SR ↑	**0.46**	0.36	0.34	0.43	0.38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Rwd ↑	-899.0	-626.2	-607.4	-695.3	-657.5	–	–	–	–	–	–	–	–
Inf-Time ↓	21.4	3.96	4.66	3.06	1.68	–	–	–	–	–	–	–	–
Shelf3D (Dynamic3D)													
SR ↑	**1.00**	0.55	0.52	0.00	0.00	0.00	0.02	0.00	0.09	0.13	0.00	0.00	0.00
Rwd ↑	-99.9	-105.9	-110.0	–	–	–	-342.7	–	-356.0	-364.7	–	–	–
Inf-Time ↓	1.59	3.45	3.68	–	–	–	3.90	–	2.63	2.52	–	–	–
SweepIntoDrawer3D (Dynamic3D)													
SR ↑	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	**0.14**	0.00	0.00	0.00
Rwd ↑	-157.4	–	–	–	–	–	–	–	-516.6	-520.6	–	–	–
Inf-Time ↓	28.3	–	–	–	–	–	–	–	3.59	3.13	–	–	–

TABLE II: Benchmark evaluations across representative Kinematic2D, Dynamic2D, Kinematic3D, and Dynamic3D environments.

Subtask	Open Drawer	Grasp Sweeper	Sweep Some Objects	Sweep All Objects
DPES	0.50	0.28	0.08	0.04
DP	0.87	0.74	0.22	0.14
VLA	0.01	0.00	0.00	0.00

TABLE III: Subtask success rate for DPES, DP, VLA baselines in the SweepIntoDrawer3D environment.

Baselines	DP				VLA			
Num. Obstacles	1	0	2	3	1	0	2	3
SR ↑	0.33	0.35	0.30	0.26	0.50	0.52	0.43	0.40

TABLE IV: DynObstruction2D out-of-distribution generalization. Training has 1 obstacle; evaluation has 0-3 obstacles.

Num. Buttons	1	3	5	10
SR ↑	0.99	0.26	0.02	0.00
Rwd ↑	-52.60	-86.30	-123.80	–
Inf-Time ↓	1.51	19.45	28.12	38.39

TABLE V: Test-time generalization for bilevel planning baseline in the StickButton2D environment.

We use an overhead camera to localize the robot and obtain object bounding boxes and poses using [105]. Given the estimated robot and object poses, we initialize the robot states and object-centric states accordingly. We then generate a plan in the simulator and execute it back in the real world. See Appendix D for additional discussion.

VIII. Limitations and Discussion

We conclude by acknowledging several limitations of the present work. First, as with any simulation-based benchmark, certain aspects of real-world physics and interaction are not fully captured. While the challenges emphasized here primarily target "mid-level" reasoning, where fine-grained physical details may be less critical, there exist important dimensions of physical reasoning for which real-world fidelity plays a more substantial role. Second, to maintain a manageable scope, we made a number of design choices that necessarily exclude other factors relevant to robotics and physical reasoning, including stochasticity, partial observability, diverse robot embodiments, and multi-robot coordination. Third, although we selected baseline methods that are standard and broadly representative, many alternative approaches were not evaluated. We look forward to actively supporting the community and maintaining KinDER open-source as researchers develop and evaluate more sophisticated methods.

ACKNOWLEDGEMENT

We acknowledge the support of the Air Force Research Laboratory (AFRL), DARPA, under agreement number FA8750-23-2-1015. We also acknowledge the Defense Science and Technology Agency (DSTA) under contract #DST000EC124000205. The authors would also like to express sincere gratitude to Dr. Caelan Garrett (NVIDIA), Dr. Shuangyu Xie (UC Berkeley), Dr. Kaiyuan Chen (UC Berkeley), Prof. David Held (CMU), Prof. Zak Kingston (Purdue University), Dr. Ajay Mandlekar (NVIDIA), Prof. Ben Eysenbach (Princeton), Prof. Jeannette Bohg (Stanford), Prof. Rika Antonova (University of Cambridge), Carlota Parés-Morlans (Stanford), Yishu Li (CMU), Prof. Florian Shkurti (University of Toronto), Prof. Greg Stein (George Mason University), and Prof. George Konidaris (Brown University) for their insightful suggestions and comments when this project was at an early stage. We thank Google's TPU Research Cloud (TRC) for providing access to Cloud TPUs.

REFERENCES

[1] E. Davis, "Physical reasoning," *Foundations of Artificial Intelligence*, vol. 3, pp. 597–620, 2008.

[2] M. Toussaint, J.-S. Ha, and D. Driess, "Describing physics for physical reasoning: Force-based sequential manipulation planning," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6209–6216, 2020.

[3] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, "Integrated task and motion planning," *Annual review of control, robotics, and autonomous systems*, vol. 4, no. 1, pp. 265–293, 2021.

[4] Y. Wang, J. Duan, D. Fox, and S. Srinivasa, "Newton: Are large language models capable of physical reasoning?" in *Findings of the association for computational linguistics: EMNLP 2023*, 2023, pp. 9743–9758.

[5] A. Cherian, R. Corcodel, S. Jain, and D. Romeres, "Llmphy: Complex physical reasoning using large language models and world models," *arXiv preprint arXiv:2411.08027*, 2024.

[6] J. Gao, B. Sarkar, F. Xia, T. Xiao, J. Wu, B. Ichter, A. Majumdar, and D. Sadigh, "Physically grounded vision-language models for robotic manipulation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 12 462–12 469.

[7] Q. Zhao, Y. Lu, M. J. Kim, Z. Fu, Z. Zhang, Y. Wu, Z. Li, Q. Ma, S. Han, C. Finn *et al.*, "Cot-vla: Visual chain-of-thought reasoning for vision-language-action models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 1702–1713.

[8] A. Ye, Z. Zhang, B. Wang, X. Wang, D. Zhang, and Z. Zhu, "Vla-r1: Enhancing reasoning in vision-language-action models," *arXiv preprint arXiv:2510.01623*, 2025.

[9] K. R. Allen, K. A. Smith, and J. B. Tenenbaum, "Rapid trial-and-error learning with simulation supports flexible tool use and physical reasoning," *Proceedings of the National Academy of Sciences*, vol. 117, no. 47, pp. 29 302–29 310, 2020.

[10] Z. Zhao, S. Cheng, Y. Ding, Z. Zhou, S. Zhang, D. Xu, and Y. Zhao, "A survey of optimization-based task and motion planning: From classical to learning approaches," *IEEE/ASME Transactions on Mechatronics*, 2024.

[11] T. L. Griffiths, F. Callaway, M. B. Chang, E. Grant, P. M. Krueger, and F. Lieder, "Doing more with less: meta-reasoning and meta-learning in humans and machines," *Current Opinion in Behavioral Sciences*, vol. 29, pp. 24–30, 2019.

[12] Y. Sung, L. P. Kaelbling, and T. Lozano-Pérez, "Learning when to quit: meta-reasoning for motion planning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4692–4699.

[13] S. Vats, M. Likhachev, and O. Kroemer, "Efficient recovery learning using model predictive meta-reasoning," *arXiv preprint arXiv:2209.13605*, 2022.

[14] M. Deitke, E. VanderBilt, A. Herrasti, L. Weihs, K. Ehsani, J. Salvador, W. Han, E. Kolve, A. Kembhavi, and R. Mottaghi, "Procthor: Large-scale embodied ai using procedural generation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5982–5994, 2022.

[15] L. Wang, Y. Ling, Z. Yuan, M. Shridhar, C. Bao, Y. Qin, B. Wang, H. Xu, and X. Wang, "Gensim: Generating robotic simulation tasks via large language models," *arXiv preprint arXiv:2310.01361*, 2023.

[16] C. Eppner, A. Murali, C. Garrett, R. O'Flaherty, T. Hermans, W. Yang, and D. Fox, "scene_synthesizer: A python library for procedural scene generation in robot manipulation," *Journal of Open Source Software*, vol. 10, no. 105, p. 7561, 2025.

[17] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, W. Ai, B. J. Martinez, H. Yin, M. Lingelbach, M. Hwang, A. Hiranaka, S. Garlanka, A. Aydin, S. Lee, J. Sun, M. Anvari, M. Sharma, D. Bansal, S. Hunter, K.-Y. Kim, A. Lou, C. R. Matthews, I. Villa-Renteria, J. H. Tang, C. Tang, F. Xia, Y. Li, S. Savarese, H. Gweon, C. K. Liu, J. Wu, and L. Fei-Fei, "Behavior-1k: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation," *arXiv preprint arXiv:2403.09227*, 2024.

[18] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, "Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks," *arXiv preprint arXiv:2112.03227*, 2021.

[19] Y. Li, Y. Zeng, Z. Huang, R. Luo, X. He, C. K. Liu, and Y. Zhu, "Dittogym: Learning to control soft robots with differentiable simulation," *arXiv preprint arXiv:2406.12452*, 2024.

[20] Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. de Las Casas, D. Budden, A. Abdolmaleki, J. Merel,

A. Lefrancq, T. Lillicrap, and M. Riedmiller, "Deepmind control suite," *arXiv preprint arXiv:1801.00690*, 2018.

[21] M. Li, S. Zhao, Q. Wang, K. Wang, Y. Zhou, S. Srivastava, C. Gokmen, T. Lee, L. E. Li, R. Zhang, W. Liu, P. Liang, L. Fei-Fei, J. Mao, and J. Wu, "Embodied agent interface: Benchmarking llms for embodied decision making," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[22] R. Yang, H. Chen, J. Zhang, M. Zhao, C. Qian, K. Wang, Q. Wang, T. V. Koripella, M. Movahedi, M. Li, H. Ji, H. Zhang, and T. Zhang, "Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents," in *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

[23] M. Heo, Y. Lee, D. Lee, and J. J. Lim, "Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation," in *Robotics: Science and Systems (RSS)*, 2023.

[24] S. Li, K. Wu, C. Zhang, and Y. Zhu, "I-phyre: Interactive physical reasoning," in *International Conference on Learning Representations (ICLR)*, 2024.

[25] M. Matthews, M. Beukman, C. Lu, and J. Foerster, "Kinetix: Investigating the training of general agents through open-ended physics-based control tasks," in *OpenReview*, 2024.

[26] F. Lagriffoul, N. T. Dantam, C. Garrett, A. Akbari, S. Srivastava, and L. E. Kavraki, "Platform-independent benchmarks for task and motion planning," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3765–3772, 2018.

[27] X. Li, W. Zhang, X. Xu, Y. Li, and Y. Zhu, "Libero: Benchmarking knowledge transfer for lifelong robot learning," in *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[28] A. Shukla, S. Tao, and H. Su, "Maniskill-hab: A benchmark for low-level manipulation in home rearrangement tasks," in *International Conference on Learning Representations (ICLR)*, 2025.

[29] S. Park, K. Frans, B. Eysenbach, and S. Levine, "Ogbench: Benchmarking offline goal-conditioned rl," in *International Conference on Learning Representations (ICLR)*, 2025.

[30] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu, "Robocasa: Large-scale simulation of household tasks for generalist robots," in *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[31] S. Zhang, Z. Xu, P. Liu, X. Yu, Y. Li, Q. Gao, Z. Fei, Z. Yin, Z. Wu, Y.-G. Jiang, and X. Qiu, "Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[32] G. J. Gao, T. Li, J. Shi, Y. Li, Z. Zhang, N. Figueroa, and D. Jayaraman, "Vlmgineer: Vision language models as robotic toolsmiths," 2025.

[33] L. P. Kaelbling and T. Lozano-Pérez, "Integrated task and motion planning in belief space," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1194–1227, 2013.

[34] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel, "Combined task and motion planning through an extensible planner-independent interface layer," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 639–646.

[35] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, "Pddlstream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning," in *Proceedings of the international conference on automated planning and scheduling*, vol. 30, 2020, pp. 440–448.

[36] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[37] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel *et al.*, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.

[38] M. Mirza, A. Jaegle, J. J. Hunt, A. Guez, S. Tunyasuvunakool, A. Muldal, T. Weber, P. Karkus, S. Racaniere, L. Buesing *et al.*, "Physically embedded planning problems: New challenges for reinforcement learning," *arXiv preprint arXiv:2009.05524*, 2020.

[39] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters, "An algorithmic perspective on imitation learning," *Foundations and Trends® in Robotics*, vol. 7, no. 1-2, pp. 1–179, 2018.

[40] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "Bc-z: Zero-shot task generalization with robotic imitation learning," in *Conference on Robot Learning*. PMLR, 2022.

[41] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, vol. 44, no. 10-11, pp. 1684–1704, 2025.

[42] S. Wang, M. Han, Z. Jiao, Z. Zhang, Y. N. Wu, S.-C. Zhu, and H. Liu, "Llmˆ 3: Large language model-based task and motion planning with motion failure reasoning," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 12 086–12 092.

[43] Y. Hu, F. Lin, T. Zhang, L. Yi, and Y. Gao, "Look before you leap: Unveiling the power of gpt-4v in robotic vision-language planning," *arXiv preprint arXiv:2311.17842*, 2023.

[44] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu *et al.*, "Palm-e: An embodied multimodal language

model," *arXiv preprint arXiv:2303.03378*, 2023.

[45] G. R. Team, A. Abdolmaleki, S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, A. Balakrishna, N. Batchelor, A. Bewley, J. Bingham *et al.*, "Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer," *arXiv preprint arXiv:2510.03342*, 2025.

[46] N. Gkanatsios, A. Jain, Z. Xian, Y. Zhang, C. G. Atkeson, and K. Fragkiadaki, "Energy-based models are zero-shot planners for compositional scene rearrangement," in *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[47] V. Jain, Y. Lin, E. Undersander, Y. Bisk, and A. Rai, "Transformers are adaptable task planners," in *Conference on Robot Learning*. PMLR, 2023, pp. 1011–1037.

[48] G. S. Cooper, "A semantic analysis of english locative prepositions," Bolt Beranek and Newman, Cambridge, MA, Tech. Rep. 1587, 1968.

[49] A. Bicchi, "Force distribution in multiple whole-limb manipulation," in *[1993] Proceedings IEEE International Conference on Robotics and Automation*. IEEE, 1993, pp. 196–201.

[50] X. Xu, D. Bauer, and S. Song, "Robopanoptes: The all-seeing robot with whole-body dexterity," *arXiv preprint arXiv:2501.05420*, 2025.

[51] Y. I. Liu, B. Li, B. Eysenbach, and T. Silver, "Slap: Shortcut learning for abstract planning," *arXiv preprint arXiv:2511.01107*, 2025.

[52] R. Madan, J. Lin, M. Goel, A. Li, A. Xie, X. Liang, M. Lee, J. Guo, P. N. Thakkar, R. Banerjee, J. Barreiros, K. Tsui, T. Silver, and T. Bhattacharjee, "Prioritouch: Adapting to user contact preferences for whole-arm physical human-robot interaction," in *Proceedings of the 9th Conference on Robot Learning (CoRL)*, 2025.

[53] J. Brüdigam, A. A. Abbas, M. Sorokin, K. Fang, B. Hung, M. Guru, S. Sosnowski, J. Wang, S. Hirche, and S. Le Cleac'h, "Jacta: A versatile planner for learning dexterous and whole-body manipulation," *arXiv preprint arXiv:2408.01258*, 2024.

[54] P. Samtani, F. Leiva, and J. Ruiz-del Solar, "Learning to break rocks with deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1077–1084, 2023.

[55] R. Holladay, T. Lozano-Pérez, and A. Rodriguez, "Force-and-motion constrained planning for tool use," in *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2019.

[56] M. A. Toussaint, K. R. Allen, K. A. Smith, and J. B. Tenenbaum, "Differentiable physics and stable modes for tool-use and manipulation planning," 2018.

[57] T. Silver, A. Athalye, J. B. Tenenbaum, T. Lozano-Pérez, and L. P. Kaelbling, "Learning neuro-symbolic skills for bilevel planning," in *Proceedings of The 6th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Tan, M. Toussaint, and K. Darvish, Eds., vol. 205. PMLR, 2023, pp. 701–714.

[58] A. Curtis, T. Silver, J. B. Tenenbaum, T. Lozano-Pérez, and L. Kaelbling, "Discovering state and action abstractions for generalized task and motion planning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, no. 5, 2022, pp. 5377–5384.

[59] F. Wang and K. Hauser, "Stable bin packing of non-convex 3d objects with a robot manipulator," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8698–8704.

[60] N. Kumar, T. Silver, W. McClinton, L. Zhao, S. Proulx, T. Lozano-Pérez, L. P. Kaelbling, and J. Barry, "Practice makes perfect: Planning to learn skill parameter policies," in *Robotics: Science and Systems (RSS)*, 2024.

[61] L. Nair, J. Balloch, and S. Chernova, "Tool macgyvering: Tool construction using geometric reasoning," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 5837–5843.

[62] T. Nickel, R. Bormann, and K. O. Arras, "Multi-heuristic robotic bin packing of regular and irregular objects," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 10 730–10 736.

[63] C. Nam, S. H. Cheong, J. Lee, D. H. Kim, and C. Kim, "Fast and resilient manipulation planning for object retrieval in cluttered and confined environments," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1539–1552, 2021.

[64] M. Stilman and J. J. Kuffner, "Navigation among movable obstacles: Real-time reasoning in complex environments," *International Journal of Humanoid Robotics*, vol. 2, no. 04, pp. 479–503, 2005.

[65] J. Ichnowski, Y. Avigal, Y. Liu, and K. Goldberg, "Gomp-fit: Grasp-optimized motion planning for fast inertial transport," in *2022 international conference on robotics and automation (ICRA)*. IEEE, 2022, pp. 5255–5261.

[66] Y. Wang, T.-H. Wang, J. Mao, M. Hagenow, and J. Shah, "Grounding language plans in demonstrations through counterfactual perturbations," *arXiv preprint arXiv:2403.17124*, 2024.

[67] L. Liu and J. Hodgins, "Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning," *Acm transactions on graphics (tog)*, vol. 37, no. 4, pp. 1–14, 2018.

[68] A. A. Rizzi and D. E. Koditschek, "Progress in spatial robot juggling." in *1992 IEEE International Conference on Robotics and Automation (ICRA)*, 1992, pp. 775–780.

[69] V. Saxena, M. Bronars, N. R. Arachchige, K. Wang, W. C. Shin, S. Nasiriany, A. Mandlekar, and D. Xu, "What matters in learning from large-scale datasets for robot manipulation," in *The Thirteenth International Conference on Learning Representations*, 2025.

[70] N. Chernyadev, N. Backshall, X. Ma, Y. Lu, Y. Seo, and S. James, "Bigym: A demo-driven mobile bi-manual manipulation benchmark," *arXiv preprint*

arXiv:2407.07788, 2024.

[71] M. Moll, I. A. Sucan, and L. E. Kavraki, "Benchmarking motion planning algorithms: An extensible infrastructure for analysis and visualization," *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 96–102, 2015.

[72] E. Heiden, L. Palmieri, L. Bruns, K. O. Arras, G. S. Sukhatme, and S. Koenig, "Bench-mr: A motion planning benchmark for wheeled mobile robots," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4536–4543, 2021.

[73] C. Chamzas, C. Quintero-Pena, Z. Kingston, A. Orthey, D. Rakita, M. Gleicher, M. Toussaint, and L. E. Kavraki, "Motionbenchmaker: A tool to generate and benchmark motion planning datasets," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 882–889, 2021.

[74] D. Long and M. Fox, "The 3rd international planning competition: Results and analysis," *Journal of Artificial Intelligence Research*, vol. 20, pp. 1–59, 2003.

[75] M. Vallati, L. Chrpa, M. Grześ, T. L. McCluskey, M. Roberts, S. Sanner *et al.*, "The 2014 international planning competition: Progress and trends," *Ai Magazine*, vol. 36, no. 3, pp. 90–98, 2015.

[76] A. Taitler, R. Alford, J. Espasa, G. Behnke, D. Fišer, M. Gimelfarb, F. Pommerening, S. Sanner, E. Scala, D. Schreiber *et al.*, "The 2023 international planning competition," 2024.

[77] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, "Alfred: A benchmark for interpreting grounded instructions for everyday tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[78] E. Kolve, R. Mottaghi, D. Gordon, Y. Zhu, A. Gupta, A. Farhadi, and L. Fei-Fei, "Ai2-thor: An interactive 3d environment for visual AI," *arXiv preprint arXiv:1712.05474*, 2017.

[79] J. Li, E. Kolve, D. Ramanan, A. Gupta, A. Farhadi, and R. Mottaghi, "Manipulathor: A framework for visual object manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[80] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. S. Chaplot, O. Maksymets *et al.*, "Habitat 2.0: Training home assistants to rearrange their habitat," *Advances in neural information processing systems*, vol. 34, pp. 251–266, 2021.

[81] W. Li, G. Chen, T. Zhao, J. Wang, T. Hu, Y. Liao, W. Guo, and S. Yuan, "Cleanupbench: Embodied sweeping and grasping benchmark," *arXiv preprint arXiv:2508.05543*, 2025.

[82] M. Cai, X. Chen, Y. An, J. Zhang, X. Wang, W. Xu, W. Zhang, and T. Liu, "Cookbench: A long-horizon embodied planning benchmark for complex cooking scenarios," *arXiv preprint arXiv:2508.03232*, 2025.

[83] A. Bakhtin, L. van der Maaten, J. Johnson, L. Gustafson, and R. Girshick, "Phyre: A new benchmark for physical reasoning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[84] A. Melnik, R. Schiewer, M. Lange, A. Muresanu, M. Saeidi, A. Garg, and H. Ritter, "Benchmarks for physical reasoning ai," *Transactions on Machine Learning Research (TMLR)*, 2023.

[85] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *Behavioral and brain sciences*, vol. 40, p. e253, 2017.

[86] M. Towers, A. Kwiatkowski, J. Terry, J. U. Balis, G. De Cola, T. Deleu, M. Goulão, A. Kallinteris, M. Krimmel, A. KG *et al.*, "Gymnasium: A standard interface for reinforcement learning environments," *arXiv preprint arXiv:2407.17032*, 2024.

[87] V. Blomqvist, "Pymunk: An easy-to-use pythonic 2d physics library," accessed 2026-01-30. [Online]. Available: https://www.pymunk.org/

[88] E. Coumans, "Bullet physics simulation," in *ACM SIGGRAPH 2015 Courses*, 2015, p. 1.

[89] J. Wu, W. Chong, R. Holmberg, A. Prasad, Y. Gao, O. Khatib, S. Song, S. Rusinkiewicz, and J. Bohg, "Tidybot++: An open-source holonomic mobile manipulator for robot learning," *arXiv preprint arXiv:2412.10447*, 2024.

[90] "Kinova gen3 7dof robotic arm," 2025, (Accessed: 1st January, 2025). [Online]. Available: https://www.kinovarobotics.com/uploads/User-Guide-Gen3-R07.pdf

[91] "Robotiq 2f-85 gripper," 2025, (Accessed: 1st January, 2025). [Online]. Available: https://robotiq.com/products/2f85-140-adaptive-robot-gripper

[92] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[93] R. S. Sutton, D. Precup, and S. Singh, "Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning," *Artificial intelligence*, 1999.

[94] D. McDermott, M. Ghallab, A. E. Howe, C. A. Knoblock, A. Ram, M. M. Veloso, D. S. Weld, and D. E. Wilkins, "PDDL-the Planning Domain Definition Language," 1998.

[95] T. Silver, R. Chitnis, N. Kumar, W. McClinton, T. Lozano-Pérez, L. Kaelbling, and J. B. Tenenbaum, "Predicate Invention for Bilevel Planning," in *Proceedings of The AAAI Conference on Artificial Intelligence (AAAI)*, vol. 37, 2023, pp. 12 120–12 129.

[96] B. Li, T. Silver, S. Scherer, and A. Gray, "Bilevel Learning for Bilevel Planning," in *Proceedings of the Robotics: Science And Systems (RSS)*, 2025.

[97] C. Agia, K. M. Jatavallabhula, M. Khodeir, O. Miksik, V. Vineet, M. Mukadam, L. Paull, and F. Shkurti, "Taskography: Evaluating robot task planning over large

3d scene graphs," in *Conference on Robot Learning*. PMLR, 2022, pp. 46–58.

[98] T. Silver, V. Hariprasad, R. S. Shuttleworth, N. Kumar, T. Lozano-Pérez, and L. P. Kaelbling, "Pddl planning with pretrained large language models," in *NeurIPS 2022 foundation models for decision making workshop*, 2022.

[99] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su, "Llm-planner: Few-shot grounded planning for embodied agents with large language models," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 2998–3009.

[100] A. Singh, A. Fry, A. Perelman, A. Tart, A. Ganesh, A. El-Kishky, A. McLaughlin, A. Low, A. Ostrow, A. Ananthram *et al.*, "Openai gpt-5 system card," *arXiv preprint arXiv:2601.03267*, 2025.

[101] T. Howell, N. Gileadi, S. Tunyasuvunakool, K. Zakka, T. Erez, and Y. Tassa, "Predictive sampling: Real-time behaviour synthesis with mujoco," *arXiv preprint arXiv:2212.00541*, 2022.

[102] R. Chitnis, T. Silver, J. B. Tenenbaum, T. Lozano-Perez, and L. P. Kaelbling, "Learning neuro-symbolic relational transition models for bilevel planning," in *2022 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2022, pp. 4166–4173.

[103] U. A. Mishra, S. Xue, Y. Chen, and D. Xu, "Generative Skill Chaining: Long-horizon Skill Planning with Diffusion Models," in *Conference on Robot Learning*. PMLR, 2023, pp. 2905–2925.

[104] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai *et al.*, "$\pi - 0.5$: A vision-language-action model with open-world generalization," *arXiv preprint arXiv:2504.16054*, 2025.

[105] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui, "Open-vocabulary object detection via vision and language knowledge distillation," in *International Conference on Learning Representations*, 2022.

[106] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Networks*, 2018.