Physics Representation Learning for Dexterous Manipulation Planning

Yi Wu[1,2], Mengsha Hu[3], Runxiang Jin[2], and Rui Liu[2*]

***Abstract*— Dexterous manipulation in robotics, particularly with high degrees of freedom (DoF) devices like the 24-joint Shadow Hand, confronts complexities in search space and execution precision. Humans, however, manipulate objects effortlessly, thanks to their innate grasp of physics. This paper introduces the Physics Representation Learning (PRL) framework for robotic hand manipulation, inspired by human dexterity. PRL uses physical principles for action conceptualization, such as aligning axes at specific angles, thereby enhancing learning efficiency and integrating action planning with execution via inverse kinematics. Implemented in a reinforcement learning network with demonstrations, PRL's effectiveness was validated on the Nvidia Isaac Sim across tasks like hammering and unscrewing. Results show PRL significantly outperforms conventional joint-control algorithms and reinforcement learning without demonstrations, demonstrating the advantages of combining physics-based action representations with expert demonstrations for complex tasks.**

I. INTRODUCTION

Recent advancements in algorithm and sensor technologies have significantly improved robotic manipulation precision and capabilities, affecting sectors ranging from everyday services and industrial assembly to medical surgeries [1], [2]. Dexterous robotic hands, essential for complex tasks like tool use, are designed to manipulate tools intended for humans, introducing a substantial challenge for learning-based methods. The high Degrees of Freedom (DoFs) in multi-finger robot hands expand the Reinforcement Learning (RL) action space, complicating joint-level training. Imitation learning emerges as a promising strategy, accelerating the learning process and producing more human-like movements [3]. Nonetheless, challenges such as biases in joint angle predictions can severely compromise grasp performance. Humans, in contrast, effortlessly manipulate objects with varied characteristics, such as geometry, size, stiffness, and weight, excelling in tool use despite external disturbances [4], [5]. This human adaptability and versatility derive from an in-depth understanding of the physics behind hand-tool-object interactions, focusing on spatial constraints and kinematic goals rather than joint-level controls.

Drawing inspiration from humans, our paper introduces the novel Physics Representation Learning (PRL) framework, aiming to replicate the human-like understanding of physics in robotic systems. This approach models manipulation actions based on physical laws (Figure 1) and redefines

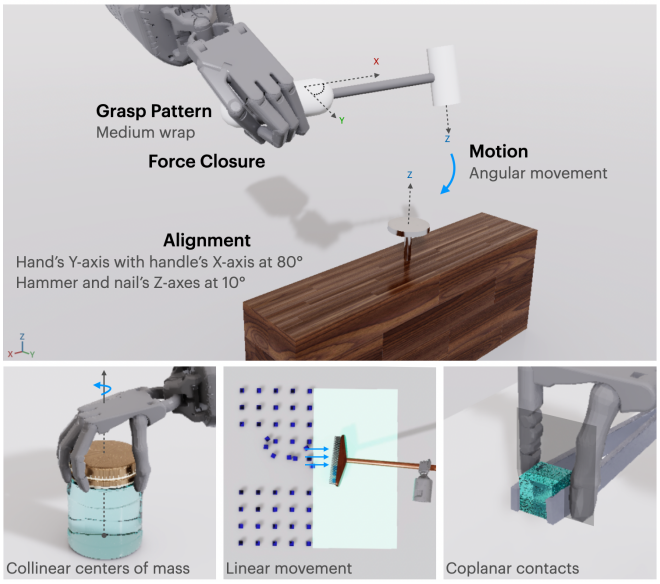

Fig. 1. Translation of physical principles into representations of action for dexterous manipulation tasks.

manipulation plans as sequences of these laws (Figure 2). The PRL framework seeks to enable robots to learn physical rules from human demonstrations and to use these rules for planning manipulations. As detailed in Figure 4, the framework starts with a Representation Network (RN) that classifies the physical principles observed in human demonstrations. Then, a Learning Network (LN) uses the physics knowledge extracted by the RN to initiate its Reinforcement Learning process, effectively pre-training the LN with data from the RN. Both the RN's physics-informed actions and the LN's action space are unified by the Physical Laws Bank (PLB) (Figure 5), a database that organizes and categorizes discretized physics-informed actions. This methodology aims to improve training efficiency by leveraging physics-based manipulation actions and insights from demonstrations for better manipulation planning.

In summary, this work has three primary contributions:

- A Physical Laws Bank (PLB) categorizes manipulation physics systematically and acts as an extensive toolkit for Embodied Agents to use for planning.
- A Physics Representation Learning (PRL) framework that implements Reinforcement Learning with demonstrations with a novel physics-guided action space.
- Four simulated manipulation tasks: hammering a nail, sweeping with a broom, unscrewing a bottle cap, and pinching a cube with tweezers are constructed in the Isaac Sim to showcase the efficiency, effectiveness, and explainability of our approach.

[1] is with the Informatique et Sciences du Numérique (Computer Research Laboratory), CentraleSupélec, Paris-Saclay University, 91190 Gif-sur-Yvette, France. [2] is with the Cognitive Robotics and AI Lab (CRAI), College of Aeronautics and Engineering; [3] is with Department of Computer Science; Kent State University, Kent, OH 44240, USA. * Rui Liu is the corresponding author, email: ruiliu.robotics@gmail.com.

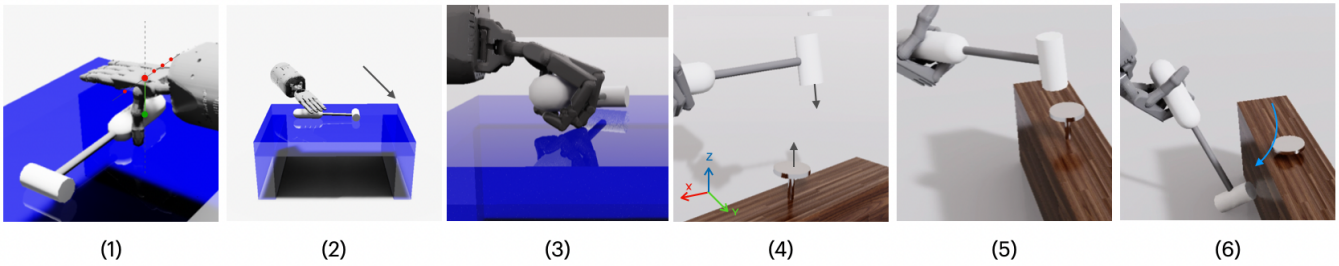

Fig. 2. **Physics-Guided Manipulation Planning.** The hammering task's manipulation plan is comprised of the following stages: **Pre-Grasping Stage:** (1) Hand-tool Positioning: Align the hammer handle's mass center with an anchor point (see Figure 3) on the palm. (2) Hand-tool Rotation: Rotate the hand around the Z-axis by 20 degrees. **Grasping Stage:** (3) Grasp Pattern: Employ a medium wrap pose to secure the handle. **Post-Grasping Stage:** (4) Tool-object Rotation: Align the hammer head with the nail along their Z-axes. (5) Tool-object Positioning: Position the hammer head approximately 3cm above the nail head. (6) Hand Movement: Perform an angular movement at the wrist joint to drive the hammer onto the nail.

II. RELATED WORK

Physics-Guided Robotics Manipulations. Traditional analytical methods define grasp quality metrics, such as form/force closure and wrench volume, to mathematically derive optimal grasping strategies, evaluating aspects like resistance, dexterity, and stability [6], [7]. These approaches, however, struggle with the computational demands of high-DoF multi-finger robot hands. Physics-guided learning methods blend traditional physics with data-driven techniques . Most physics-guided learning methods integrate these grasping metrics into the RL reward functions, with examples including force-closure quality indices [8] and hierarchical rewards incorporating grasping matrix and wrench volume metrics [9]. Another approach saw the integration of Dynamic Movement Primitives (DMPs) directly into the policy network to generate desired actions, albeit still facing computational inefficiencies [10]. Our method diverges by reframing the RL action space itself, representing dexterous manipulation actions through underlying physical laws. This innovative approach significantly enhances learning efficiency and broadens the applicability of robotic manipulation.

Reinforcement Learning with Demonstrations. Incorporating human demonstrations into robot training accelerates the learning process, enabling robots to mimic natural, efficient human movements. Research has increasingly focused on enhancing RL with demonstrations to improve learning efficiency [11]–[14]. A notable example by [15] uses demonstrations for initial policy setup via Behavioral Cloning (BC) and refines this with RL, adding penalties for deviating from demonstrated actions. However, BC focuses solely on the policy network, and overlooks the value network and reward signals, potentially limiting effectiveness. In contrast, [13] boosts the DDPG [16] method by incorporating demonstrations directly into the replay buffer, enriching the training dataset without extra interactions. Drawing inspiration from these approaches, we innovate by pre-training the on-policy A2C [17] algorithm with expert demonstrations. This not only steers the network closer to expert performance from the outset but also overcomes the exploration limitations associated with BC.

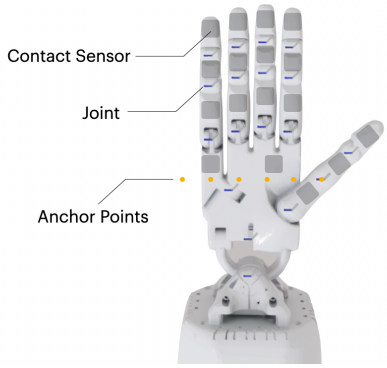

Fig. 3. **Shadow Hand Model. Joints:** 30 joints in total—22 in fingers, 2 in wrist, plus 6 for arm control, enabling movements along and rotations around X, Y, and Z axes. **Sensors:** In the palm's grey area, detecting contact information as detailed in Section III. **Anchor Points:** Palm anchors for tool positional alignment.

III. PROBLEM STATEMENT

We focus on a sophisticated tool-use manipulation task, symbolized by $\mathcal{M} = \langle \mathcal{H}, \mathcal{T}, \mathcal{O} \rangle$. This involves a multi-fingered robotic hand $\mathcal{H}$ manipulating a tool $\mathcal{T}$ to interact with the target object $\mathcal{O}$. The tool is characterized by two main components: a graspable volume T_{gv}, such as the handle of a hammer, and a functional volume T_{fv}, like the head of the hammer, intended for direct interaction with the object.

Figure 3 shows the robotic hand $\mathcal{H}$ equipped with 17 contact sensors ($C^i, i = 0, 1, ..., 16$) across its palm. These sensors measure contact force (C_f), position (C_{pos}), and normal (C_{norm}). The hand features 30 joints (J^i, for $i \in 0, 1, \ldots, 29$) across the arm, wrist, and fingers, with each joint tracking its velocity (J^i_{vel}) and position (J^i_{pos}). Additionally, the hand's orientation relative to the tool is indicated by six positional anchor points (A^i, where $i \in 0, 1, \ldots, 5$) on the palm, serving as stable reference points for spatial alignment.

IV. METHOD

In this section, we introduce the Physics Representation Learning (PRL) framework, integrating Reinforcement Learning with Demonstrations and novel, physics-informed action representations. At the heart of our method is the

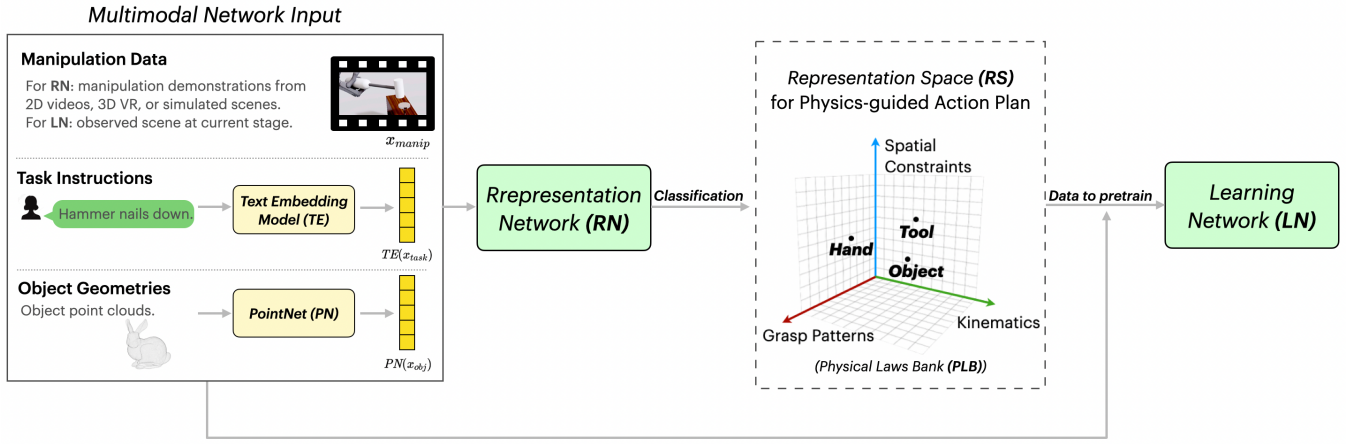

Fig. 4. **Physics Representation Learning (PRL) Framework.** The PRL Framework features two main components: **(1) Representation Network (RN):** a classifier that processes multimodal inputs from expert demonstrations, encompassing manipulation data from demonstrations, semantic task instructions, and object 3D geometries, to produce an action plan vector in the **Representation Space** (**RS**). This expert-derived network inputs and its classified outcomes are fed into **(2) Learning Network (LN):** a Reinforcement Learning network with an action space derived from the same **RS** for pretraining, which significantly enhances dexterous manipulation planning's efficiency through physics-informed actions and expert demonstrations.

Physical Laws Bank (PLB), a comprehensive database that systematically encodes physical laws for direct application in dexterous manipulation tasks. Detailed discussions on these key components are provided in the following sections.

A. Physics Laws Bank and the Rerpesentation Space

The Physics Laws Bank (PLB), illustrated in Figure 5, is a foundational database that catalogs manipulation actions as physics-based descriptions throughout the manipulation process. We break down a complete manipulation plan into three key stages: pre-grasping, which mainly addresses hand-tool spatial relationships; grasping, selecting patterns like power or precision grasp; and post-grasping, focusing on tool-object interactions and the hand's movements.

To bridge theory with practice, we discritize physical laws into concrete, selectable options. Take, for example, the precise degrees of hand-tool relative rotation, articulated as discrete angles (e.g., -180, -170, ..., 170, 180 degrees) as outlined in Table II. This method of discretization translates a manipulation plan into an integer vector v in the high-dimensional Representation Space (RS), where similar tasks or object manipulations cluster closely. Mathematically, this concept is expressed as:

$$\mathbf{RS} = \mathbf{v} \in \mathbb{Z}^n \tag{1}$$

where $\mathbf{v} = [v_1, v_2, \ldots, v_n]^\top$ denotes the integer vector corresponding to a sequence of manipulation sub-steps of physical principles or action choices, $\mathbb{Z}$ represents the set of integer numbers, and n represents the number of dimensions in this space. The discretization strikes a balance between the computational complexity and the generalization capabilities.

B. Physics Representation Learning Framework

By incorporating our physics-informed Representation Space (RS) for manipulation planning, our Physics Representation Learning (PRL) framework aims to substantially

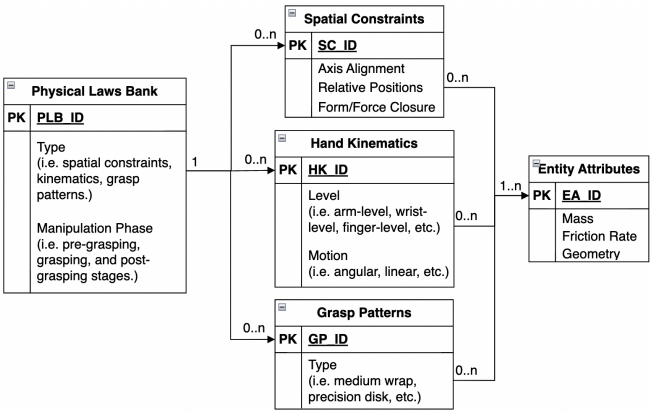

Fig. 5. **Physics Laws Bank (PLB) Structure.** The PLB is a structured database for storing and categorizing physical actions throughout the manipulation phases. It encompasses spatial constraints, hand kinematics, and grasp patterns corresponding to each phase. Each physical law is associated with one or many entities.

improve the efficiency of the learning process compared to conventional joint-level control.

Network Inputs. As illustrated in Figure 4, both the Representation Network (RN) and the Learning Network (LN) within the PRL framework process identical inputs, represented by:

$$x = [x_{\text{manip}}, TE(x_{\text{task}}), PN(x_{\text{obj}})] \tag{2}$$

Here, x_{manip} shows manipulation demonstration data, either pixel-level 2D video demonstrations, 3D VR demonstrations, or omniscient data from synthetic scenes in the simulator, such as contact forces, joint torques, object transformations and rotations, etc. x_{task} refers to the task objectives expressed in natural languages, while x_{obj} details the object's geometry. The function $TE(\cdot)$ signifies the application of text-embedding models to distill feature vectors from semantic task descriptions, and $PN(\cdot)$ denotes the PointNet model's role in encoding object point clouds into compact latent vectors via the second-last layer's output. The inclusive

network inputs significantly enhance the PRL framework's capacity to generalize across an extensive array of tasks and objects.

Representation Network (RN). The Representation Network (RN) operates as a supervised multi-classification framework that outputs the physical laws as an integer vector $\mathbf{v}$ within the RS, as illustrated in Section IV-A. It utilizes supervised labels for training, and the way to generate labels is detailed in the Appendix. After applying the Domain Randomization on network inputs that encompass various task goals and object attributes like mass, geometries, and frictionrates, and training with the supervised labels, the RN is capable of discerning the underlying physical laws applicable to novel scenarios.

The architecture of the RN network is illustrated in Table I, where len(x) indicates the length of the input vector x, and $\sum_n |dim_n|$ denotes the total number of output classes over n dimensions.

TABLE I
REPRESENTATION NETWORK (RN) ARCHITECTURE

Layer Configuration
`nn.Linear(len(x), 512), nn.BatchNorm1d(512), nn.ReLU(), nn.Dropout(p=0.5)`
`nn.Linear(512, 256), nn.BatchNorm1d(256), nn.ReLU(), nn.Dropout(p=0.5)`
`nn.Linear(256, 128), nn.BatchNorm1d(128), nn.ReLU(), nn.Dropout(p=0.4)`
`nn.Linear(128,` $\sum_n \|dim_n\|$`), nn.CrossEntropyLoss()`

Our network's objective is not just to match classification labels accurately but to uncover the key physical laws necessary for successfully completing a task. This involves identifying crucial rules that all successful action plans share, rather than classifying every detail. For example, when opening a bottle, the critical action is the relative positions, such as aligning the palm center with the bottle cap along the vertical Z-axis, whereas other physical laws, like their relative rotations around the Z-axis, are comparatively less significant for the task's success. Therefore, our RN's loss function L_{RN} is specifically designed to pinpoint these essential physical laws, focusing on what truly matters for task success, and it is denoted as:

$$
\begin{aligned}
L = -\frac{1}{N}\sum_{i=1}^{n}\sum_{k=1}^{m_i} y_{ik}\log(\hat{y}_{ik}) \\
+ \lambda \sum_{i=1}^{n} \max(0, \frac{\tau_c}{m_i} - \hat{y}_i^{\max}) \\
+ \gamma \cdot \mathbf{1}\left\{\sum_{i=1}^{n} \mathbf{1}\{\hat{y}_i^{\max} > \frac{\tau_s}{m_i}\} < T_{\min}\right\}
\end{aligned} \tag{3}
$$

where the first term is the cross-entropy loss over n classes each with m_i categories. The term y_{ik} denotes the true label, and $\hat{y}_{ik}$ represents the predicted probability for the k^{th} category within the i^{th} class. The second term introduces a penalty for predictions of the max probability category that falls below the confidence threshold $\frac{\tau_c}{m_i}$, where m_i is a normalization term, thereby incentivizing the model not only to make accurate predictions but also to be confident in its predictions. The third term imposes a penalty if the number of confident predictions that exceed the threshold $\frac{\tau_s}{m_i}$ is less than $T_{\min}$ times within a batch, thus promoting consistent confidence across predictions. The function $1\{\cdot\}$ serves as an indicator function. It returns 1 if the condition inside the braces is true, and 0 otherwise. The hyperparameters λ and γ control the influence of the confidence and consistency penalty terms, respectively.

Learning Network (LN). Our approach leverages the well-established Actor-Critic architecture (A2C [17]), diverging from the conventional joint-control action space to adopt a physics-informed action space. This choice allows for the complex task of long-duration dexterous manipulation planning to be modeled as a simple discrete integer vector v within the physics-informed Representation Space (RS). As a result, our reward design r is clear-cut and sparse, exemplified in Section V-A, aiming solely to capture the essence of task completion without the need for complex, hierarchical reward engineering typically required for joint-control manipulation planning. To enhance the learning efficiency, we pre-train the A2C network with RN's multimodal inputs x and classified physical laws. For Sim2Real transferability, we selectively utilize manipulation demonstration data x_{manip} from the initial phase instead of the entire sequence. This focused approach allows for more effective adaptation to novel environments.

V. EVALUATION

To validate the effectiveness of our proposed Physics Representation Learning (PRL) framework, we built four manipulation tasks in the Nvidia Isaac Sim simulator: hammering, sweeping, unscrewing, and tweezing, as shown in Figure 6. These scenarios were chosen to cover a wide range of grasps and motions, allowing us to demonstrate the generalization capabilities of our approach. The baseline is run with Intel i9-13900H CPU, 32 GB RAM, and NVIDIA GeForce RTX 4070 GPU.

A. Task Design

Hammering: The robot is to pick up a hammer, approach a nail, and strike it. The reward, $R_{\text{hammering}} = d$, depends on the distance (in cm) the nail is driven down. A descent of $d > 0$ cm indicates success.

Sweeping: The robot must pick up the broom, place it tail-down, and sweep right to move cubes into the reward zone. The reward for sweeping, $R_{\text{sweeping}} = N$, corresponds to the count of debris cubes relocated. A count $N > 3$ signifies a successful sweep.

Unscrewing: The robot is to unscrew the bottle cap. The reward for unscrewing, $R_{\text{unscrewing}} = \theta$, is the cap's rotation in degrees. A rotation $\theta > 8$ degrees marks success.

Tweezing: The robot must use tweezers to lift a cube. The reward for tweezing, $R_{\text{tweezing}} = l$, is the lift height in centimeters. Lifting $l > 10$ cm deems the task successful.

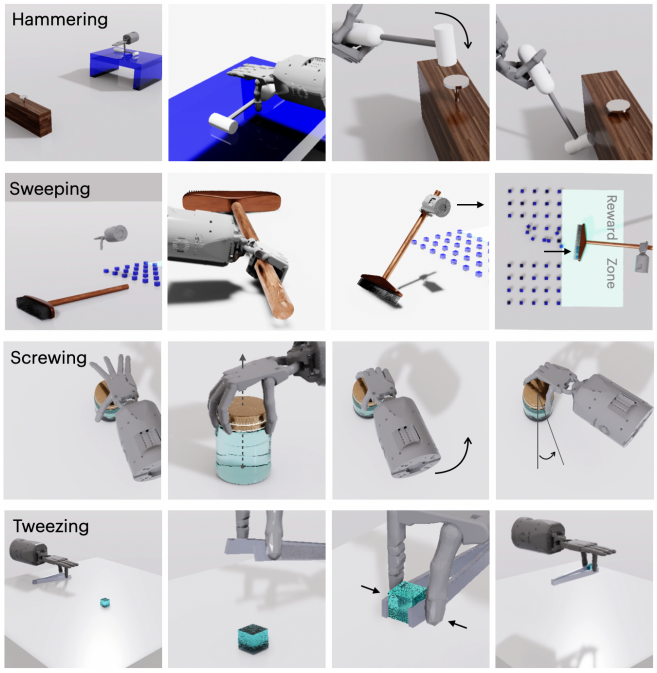

Fig. 6. **Four Simulated Tasks**. **Hammering:** The robot hand picks up a hammer, approaches a nail, and drives it down. **Sweeping:** It sweeps trash cubes into a reward zone using a broom. **Screwing:** The task involves unscrewing a bottle cap open. **Tweezing:** The hand uses tweezers to lift a cube.

B. Experimental Setup

Shadow Hand Model. In this study, the Shadow Hand model, as shown in Figure 3, serves as the multi-fingered robotic hand of choice. We have enhanced the standard model, which originally features 24 Degrees of Freedom (DoFs), by integrating an additional six arm-level joints. These additional joints include three prismatic joints that allow for translational movement along the X, Y, and Z axes in the world frame, as well as three revolute joints that enable rotational motion around the X, Y, and Z axes in the local frame. Through this augmentation, we present a hand-arm model that boasts a total of 30 DoFs, thereby advancing our research objectives.

Action Representation Space. We devised a physics-informed Representation Space (RS) with a multi-dimensional structure specified as $[7, 7, 6, 4, 6, 7]$, detailed in Table II. For example, this space includes actions like a_0^1, which identifies the action within the second category of the first dimension. This particular action corresponds to the precise adjustment of the hand-tool relative rotation around the X-axis to a negative five degrees. The creation of such an RS not only acts as a comprehensive multi-classification schema for RN but also functions as an action space for LN's Reinforcement Learning algorithms.

Training Setup for the Representation Network (RN). We gathered 8,000 successful manipulation plans via synthesizing virtual scenes in the simulator, 2,000 for each of the four tasks. The network is trained 200 epochs. To enhance training robustness, we applied Domain Randomization to include objects with various sizes and friction rates. The network's supervised labels were derived using methodologies elaborated in the Appendix. The design of our network, detailed in Table I, incorporates a loss function defined in Equation (3), with hyperparameters set to $\lambda = 1 \times bs$, $\gamma = 0.01 \times bs$, $\tau_c = \tau_s = 2$, and $T_{min} = bs/10$, where bs denotes the batch size set as 64.

TABLE II
PHYSICS-REPRESENTED ACTION PLANS IN THE EXPERIMENT

Manipulation Sub-steps	Dim	Discrete Action Choices
Hand-tool relative rotation (X-axis)	7	Degrees in [0, -5, -10, -15, -20, -25, -30]
Hand-tool relative rotation (Z-axis)	7	Degrees in [-30, -20, -10, 0, 10, 20, 30]
Hand-tool relative position	6	Six positional anchors (Figure 3) to be aligned with the center of the tool's graspable volume T_{gv} along the Z-axis, 10 cm apart.
Grasping patterns	4	Medium wrap, parallel extension, palmer pinch, and precision disk.
Hand kinematics motions	7	Wrist-level horizontal/vertical angular movement, arm-level linear movement along X, Y, Z axes, arm-level rotating around palm center, and finger-level squeezing.
Tool-object relative rotation	6	Aligning the Z-axis of the object with one of the X, Y, Z, X-Y, X-Z, or Y-Z axes of the tool's functioning volume T_{fv}.

For the network input $[x_{\text{manip}}, TE(x_{\text{task}}), PN(x_{\text{obj}})]$, as detailed in Equation (2), x_{manip} is derived directly from the simulator. It encompasses joint positions and velocities, contact reports, as well as transformations and rotations of the hand, tools, and objects, resulting in a total flattened input dimensionality of 786. Textual descriptions of tasks are converted into a 1536-dimensional vector space using OpenAI's "text-embedding-ada-002" model. Furthermore, the PointNet architecture [18] serves as our feature extractor for tool and object point-cloud data, outputting a 256-dimensional feature vector for each.

Training Setup for the Learning Network (LN). We utilize the A2C algorithm [17] as basic RL network structure, but with a physics-informed action space, as shown in Table II. Initially, we evaluate its performance against traditional joint-level control methods. Subsequently, we explore the efficacy of Pure RL versus RL augmented with expert demonstrations. Moreover, we undertake ablation studies to assess how expert models at varying degrees of development distinctly influence the acceleration of the RL training process.

The A2C algorithm is set up with the following parameters: an action spaces as illustrated in Table II, a neural network architecture with two hidden layers, each comprising 256 units, state inputs same as those used for the RN, a learning rate of 0.0007, 64 steps per update, a discount factor of 0.99, GAE Lambda of 0.95, and an entropy coefficient of 0.01. Training proceeds for 3200 timesteps, with each episode limited to 1500 physics frames, effectively making a single step encompass all 1500 frames to conclude an episode. For conventional joint-level control, the episode length is reduced to 1000 frames with a 10-frame skip per

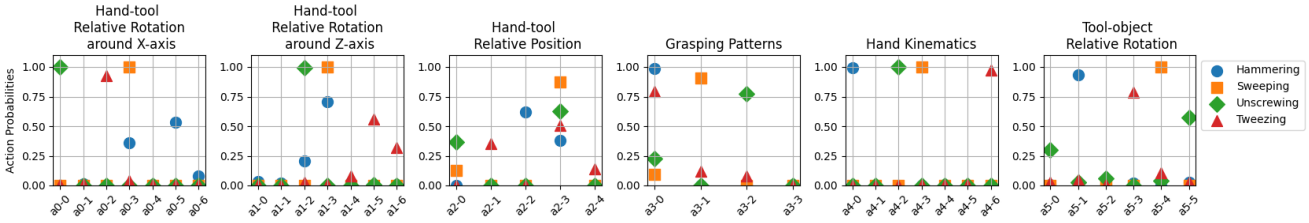

Fig. 7. **Explainable Classification Results for the Representation Network (RN)**. This figure displays RN's classification outcomes after training for 200 epochs, with action classes from a_0 to a_5 on the X-axis, as detailed in the experimental setup (refer to Table II). The Y-axis shows the predicted probabilities for each category within these classes. A high concentration of probabilities indicates key categories critical for task success, reflecting strong prediction confidence, exemplified by the distinct hand kinematics across four tasks.

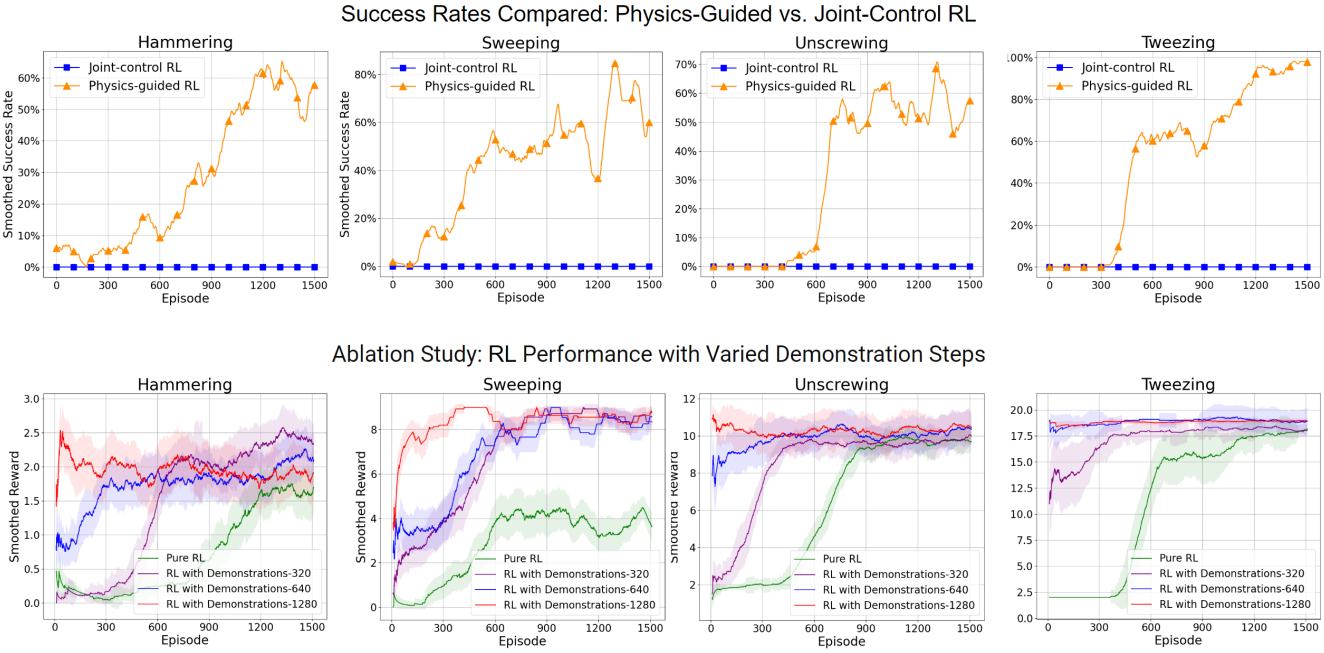

Fig. 8. **Results for the Learning Network (LN)**. The top part of the figure reveals the limitations of a joint-control RL algorithm for prolonged dexterous tasks under sparse rewards, contrasting with the superior task success rates of our physics-guided method. The bottom part demonstrates how integrating demonstrations with RL enhances learning efficiency beyond the benefits of physics-represented action space.

step, resulting in a maximum of 150 steps per episode.

C. Results Analysis

Results for the Representation Network (RN). To illustrate the effectiveness of the RN after training, we utilized the model trained up to the 200^{th} epoch. This model trained with task variety and object domain randomizations, was employed to estimate the action probabilities for our four given tasks, hammering, sweeping, unscrewing, and tweezing, based on successful expert demonstrations. The outcomes of these predictions are depicted in Figure 7. Following the experiment setup in Section V-B Action Representation Space, on the X-axis, we categorize action classes from a_0 to a_5, each comprising a distinct number of categories—for instance, a_0^0 to a_0^6, denoting the six possible hand-tool rotations around the X-axis. The Y-axis quantifies the predicted probabilities, ensuring that within a class, the sum of category probabilities equals 1. The concentration of probabilities highlights which categories are deemed vital for the success of a task, indicating high prediction confidence. Conversely, the dispersion of probability distributions across categories signals a class's relative unimportance in influencing task success.

For instance, in the hammering task, crucial factors include specific grasping patterns (medium wrap), hand kinematics (wrist-level horizontal angular movement), and the alignment of the nail's Z-axis with the hammer head's Y-axis. Similarly, for unscrewing, the relative rotation between the hand and the bottle cap is critical. These observations affirm that our physics-guided action representations not only effectively capture key physical principles but also significantly bolster the network's explainability.

Results for the Learning Network (LN). The comparative analysis between Physics-guided RL and Joint-Control RL is visualized in the upper section of Figure 8. The enhanced success rates of the Physics-guided approach are illustrated by the orange trajectories across all tasks, underscoring its ability to harness physical principles to

achieve superior task completion. On the other hand, the joint-control RL approach, indicated by the blue trajectories, demonstrates negligible improvements, struggling to adapt to the complexity of long-term dexterous manipulation tasks.

An ablation study further investigates the role of demonstration steps in RL performance. The ablation test section of Figure 8 reveals that integrating demonstrations into RL (non-green lines) yields a remarkable performance boost over pure RL methods. The pre-training with 1280 expert demonstration steps—represented by the red line—achieves superior results compared to other variants. However, the marginal benefits reduce beyond a point in the hammering task, suggesting there is an optimal number of demonstration steps that maximize performance gains while maintaining computational efficiency.

The experimental results underscore the efficacy of integrating physics-based knowledge into the RL framework. The substantial improvement in success rates brought forth by the Physics-guided RL suggests that understanding and incorporating physical interaction laws can greatly enhance the manipulation capabilities of RL agents. Moreover, the inclusion of expert demonstrations further accelerates the learning process, thereby reducing the training time required to achieve high levels of task performance.

VI. Conclusion and Future Work

This study presents the Physics Representation Learning (PRL) framework, significantly advancing dexterous manipulation with a physics-informed action space and expert demonstrations. By leveraging physical laws in action conceptualization and execution, and blending physics semantics with joint execution through inverse kinematics, PRL not only enhances learning efficiency but also paves the way for semantic-level planning with large language models (LLMs). Future work will explore the integration of this approach with advanced LLMs for broader application and a deeper understanding of robotic manipulation.

APPENDIX

A. Generating Physical Law Labels for the Representation Network (RN)

By leveraging the comprehensive data within the simulator, we introduce a computational approach to produce supervised labels for the RN. Whether synthesizing scenes for virtual demonstrations or reconstructing 3D models from 2D videos, we have access to exhaustive data for deducing the underlying physical principles. This computation presumes that the hand is in contact with the tool.

Relative Rotation Classifications. To classify hand-tool or tool-object relative rotations succinctly, we discretize their relative Euler angles $(\theta_x, \theta_y, \theta_z)$ around the X, Y, and Z axes respectively, by rounding to the nearest multiple of a specified degree constant c. For instance, $c = 10$ specifically segments the rotation into intervals of -180, -170, ..., 170, and 180 degrees. The classification Θ_{rotation} is defined as:

$$\Theta_{\text{rel\_rot}} = \left[\lfloor \theta_x/c + 0.5 \rfloor, \lfloor \theta_y/c + 0.5 \rfloor, \lfloor \theta_z/c + 0.5 \rfloor\right] \quad (4)$$

Relative Position Classifications. To assess the hand-tool relative positions, the procedure begins by locating the mass center of the tool's graspable volume, see Figure XX. Starting from this central point, six vectors ($\vec{gv}_i$) are extended outward towards the centers of the bounding box's faces, where $i \in \{0, 1, \ldots, 5\}$. Concurrently, specific co-planar anchor points are designated on the hand's surface. Drawing vectors ($\vec{h}_j$) from the tool's volume center to these anchor points allows for the evaluation of vector pair alignment ($\vec{gv}_i$, $\vec{h}_j$) through cosine similarity, determining their directional parallelism and, hence, their relative positioning. Similarly, to evaluate tool-object relative positions, the tool's functional volume is considered, projecting vectors towards volume surface centers, denoted as ($\vec{fv}_i$), and also towards the object co-planar anchor points, represented by vector $\vec{o}_j$). This assessment is encapsulated mathematically as follows:

$$\begin{aligned} S_{\text{rel\_pos\_HT}} &= \arg\max_{i,j} \cos(\vec{gv}_i, \vec{h}_j) \\ S_{\text{rel\_pos\_TO}} &= \arg\max_{i,j} \cos(\vec{fv}_i, \vec{o}_j) \end{aligned} \quad (5)$$

Grasp Pattern Classifications. To effectively classify grasp patterns, such as power and precision grasps, we leverage the data from contact sensors configured within the simulator. A total of 17 sensors are strategically placed across the palm, as depicted in Figure XX. Each sensor's state is converted into a binary variable indicating contact presence, forming a binarized contact force vector, $\vec{c_f}$, which is compared to predefined grasp pattern vectors in the Physics Laws Bank, where each grasp pattern $\vec{gp}_i$ is represented as a binary vector (e.g., the two-finger precision grasp denoted as [1,1,0,0,...]). The grasp pattern category is determined as follows:

$$G_{\text{pattern}} = \arg\max_i J(\vec{c_f}, \vec{gp}_i) \quad (6)$$

where $J(\vec{c}, \vec{gp}_i)$ is the Jaccard similarity coefficient to compare the similarity of binary vectors, mathematically denoted as:

$$J(\vec{c_f}, \vec{gp}_i) = \frac{|\vec{c_f} \cap \vec{gp}_i|}{|\vec{c_f} \cup \vec{gp}_i|} \quad (7)$$

Grasp patterns can also be inferred by examining the colinearity or coplanarity of contact points. The evaluation is succinctly captured through:

$$\begin{aligned} C_{\text{line}} &= \begin{cases} 1 & \text{if } \sigma\left(-\left|\det\left(\vec{c}_{\text{pos}}\right)\right|\right) > \tau_{\text{pos}} \\ 0 & \text{otherwise} \end{cases} \\ C_{\text{plane}} &= \begin{cases} 1 & \text{if } \sigma\left(-\left|\det\left(\vec{c}_{\text{norm}}\right)\right|\right) > \tau_{\text{norm}} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (8)$$

where σ denotes the logistic function, applied to the absolute value of the determinant (det) of the contact points' position matrix $\vec{c_{pos}}$ for collinearity and the contact normals' matrix $\vec{c_{norm}}$ for coplanarity, where a determinant value near zero indicates either perfect collinearity or coplanarity, regularized by a threshold τ_{pos} or τ_{norm}, respectively.

Kinematics Movement Classifications. Movement classifications are determined by the highest cosine similarity between the observed joint velocities, $\vec{v}_{\text{obs}}$, and predefined motion vectors, $\vec{kin}^{\,k}$, for each type of kinematic action. Formally:

$$K_{\text{mov}} = \arg\max_{k} \cos(\vec{v}_{\text{obs}}, \vec{kin}^{\,k}) \tag{9}$$

where $k \in \{0, 1, \ldots, 12\}$ indexes the 13 predefined kinematic motions, ranging from arm and wrist movements to finger manipulations.

REFERENCES

[1] F. Wei, D. Yang, Z. Luo, J. Wu, D. Su, and P. Shang, "Robust adaptive following by uwb sensors and high-precision gyroscope for industrial assistance robots," in *2022 4th International Conference on Communications, Information System and Computer Engineering (CISCE)*, pp. 572–576, 2022.

[2] C. Li and Q. Bi, "Vision-driven high precision positioning method for bracket assembly with industrial robot," in *2022 5th World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM)*, pp. 825–830, 2022.

[3] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang, "Dexmv: Imitation learning for dexterous manipulation from human videos," in *European Conference on Computer Vision*, pp. 570–587, Springer, 2022.

[4] D. Subedi, W. Jiang, R. T. Rahman, H. Zhang, K. Huang, and Y.-H. Su, "Smoothness constrained curiosity driven multicamera trajectory optimization for robot-assisted minimally invasive surgery," in *2023 International Symposium on Medical Robotics (ISMR)*, pp. 1–7, 2023.

[5] G. Rao, X. Yang, J. Xu, K. Chen, and G. Zhang, "Smoothness optimization based on measured 3d point cloud in robotic drilling," in *2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 797–802, 2017.

[6] H. Zhang, J. Tang, S. Sun, and X. Lan, "Robotic grasping from classical to modern: A survey," *arXiv preprint arXiv:2202.03631*, 2022.

[7] M. A. Roa and R. Suárez, "Grasp quality measures: review and performance," *Autonomous robots*, vol. 38, pp. 65–88, 2015.

[8] M. Monforte and F. Ficuciello, "A reinforcement learning method using multifunctional principal component analysis for human-like grasping," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 13, no. 1, pp. 132–140, 2020.

[9] Y. Jung, L. Tao, M. Bowman, J. Zhang, and X. Zhang, "Physics-guided hierarchical reward mechanism for learningbased multi-finger object grasping," *arXiv preprint arXiv:2205.13561*, 2022.

[10] S. Bahl, M. Mukadam, A. Gupta, and D. Pathak, "Neural dynamic policies for end-to-end sensorimotor learning," 2020.

[11] A. Gupta, C. Eppner, S. Levine, and P. Abbeel, "Learning dexterous manipulation for a soft robotic hand from human demonstration," 2017.

[12] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," 2016.

[13] M. Vecerik, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe, and M. Riedmiller, "Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards," 2018.

[14] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "Deepmimic: example-guided deep reinforcement learning of physics-based character skills," *ACM Transactions on Graphics*, vol. 37, p. 1–14, July 2018.

[15] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," 2018.

[16] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2019.

[17] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," 2016.

[18] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," 2017.