BEYOND VERBS AND TRAJECTORIES: PHYSICAL ACTION PRIMITIVES THROUGH LOCAL INTERACTION LAWS

Anonymous authors
Paper under double-blind review

ABSTRACT

Similar actions need not yield reusable experience: interactions with similar action labels, trajectories, and nominal outcomes can respond differently when interventions change. We introduce **Physical Action Primitives (PAPs)**, reusable action knowledge objects that preserve **local interaction laws**, how physical responses vary with intervention. A PAP retains this relationship as a persistent physical operator, supporting outcome prediction and intervention selection for new requests and contexts. Its explicit physical interface allows execution implementations to be replaced or calibrated while the response knowledge is retained. Independent physical audits in matched manipulation contexts show that PAP-based organization reduces audit discrepancy area by 26.7% relative to semantic organization and 19.1% relative to trajectory organization. Closely matched results from affine and nonlinear constructions support this organizing principle across model classes. Retained PAPs predict responses beyond their source probes, guide repeated invocation, and carry source information whose controlled exchange changes control accuracy. Real-robot experiments on a Franka Panda demonstrate prediction and invocation from real interaction evidence without real-world fine-tuning. These results establish a basis for robot experience to accumulate as libraries of persistent physical operators that planning, execution, and adaptation can access through a common interface. Code: https://anonymous.4open.science/r/PAP-B093/

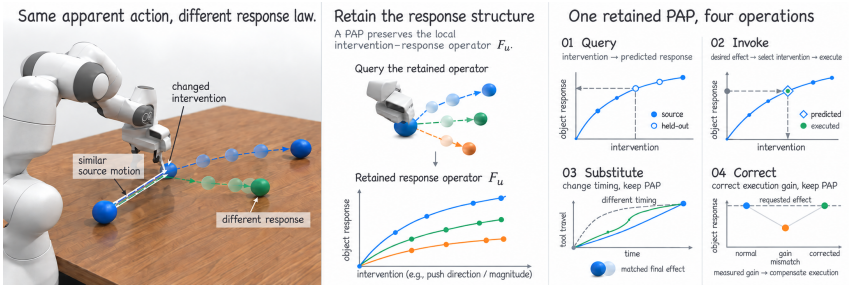


Figure 1: Physical Action Primitives (PAPs) retain local intervention–response structure as reusable action knowledge. PAPs preserve how physical responses change with intervention. The retained object supports **Query** and **Invoke**; **Substitute** and **Correct** adjust execution while preserving the response operator. Prediction and control access the same physical action knowledge across changing requests.

1 INTRODUCTION

Action primitives provide a natural way for robots to reuse experience (Sutton et al., 1999; Ijspeert et al., 2013). A robot that has already interacted with an object should be able to carry useful knowledge forward when the desired displacement, direction, velocity, or downstream request changes. This idea underlies movement primitives (Saveriano et al., 2023), reusable skills and learned action representations (Lynch et al., 2020; Pertsch et al., 2021), and affordance models (Ardón et al., 2020): experience is organized into units that can later be recalled and reused. Yet a basic question remains unresolved. **What should a reusable action primitive preserve?**

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Common ways of describing and organizing action experience draw on semantic labels (Ichter et al., 2023), motion representations (Ijspeert et al., 2013; Paraschos et al., 2013), nominal effects, or learned skill representations (Lynch et al., 2020; Pertsch et al., 2021). Each captures a meaningful aspect of behavior, but physical reuse places a different requirement on the representation. For physical reuse, a useful organizing principle is whether interactions exhibit compatible responses to changed interventions within matched contexts and a declared operating range. A shared verb or similar tool trajectory does not establish this compatibility. Two interactions can appear nearly identical in their observed motions and nominal outcomes while responding differently when the intervention changes. Once a robot receives a new physical request, this hidden distinction becomes consequential because selecting an appropriate intervention requires knowledge of how outcomes vary around the original interaction.

We propose that this reusable structure is captured by **local interaction laws**. We organize action knowledge by the local relationship between intervention and physical response. We call the resulting action objects **Physical Action Primitives (PAPs)**. A PAP retains the response structure revealed by interaction so that knowledge acquired under one request can remain useful when the desired physical effect changes. The primitive retains how nearby interventions are expected to transform physical response (Fig. 1).

This raises another question: how should this response knowledge be retained so that a robot can use it operationally? We make the response operator itself explicit. Once retained, the same PAP supports four operations. **Query** returns a predicted response in physical coordinates and units for a given context and intervention; **Invoke** uses the PAP to select and execute an intervention for a desired effect; **Substitute** preserves the PAP and request while replacing the implementation used to realize the intervention; and **Correct** uses measurements to update the execution mapping while leaving the retained PAP unchanged. Planning, execution, and monitoring can exchange desired effects, proposed interventions, predicted responses, and measured outcomes through the same retained operator.

This object definition is model-class agnostic: locally affine and nonlinear response models can instantiate PAPs. We develop a concrete construction that identifies such response operators from source interactions and exposes them through a callable physical response model. The complete operator is instantiated by its source condition, shared model, and declared physical conventions, which together determine the PAP that is retained and reused. This construction separates the physical knowledge carried by the primitive from the particular mechanism used to realize an intervention. The response knowledge can therefore persist while requests, receiving conditions, or execution implementations change, and while measured execution mappings are corrected.

Our experiments show that, in matched RollBall contexts with varying damping, organizing interactions by physical response reveals distinctions that semantic, trajectory, learned-skill, and nominal-effect organization leave unresolved. Across independently audited interventions, PAP organization produces more compatible response groups, and closely matched results from an alternative nonlinear organizer support the same organizing principle. The retained operators also predict previously unseen intervention responses and support invocation when requests and receiving conditions change. When the retained source condition is exchanged, execution accuracy changes accordingly, directly linking the stored response knowledge to subsequent control. Finally, execution substitution retains the response object across temporal profiles, and measured correction restores successful invocation on PokeCube and RollBall under a known execution-scale mismatch. These operations preserve response knowledge while modifying intervention realization.

Together, these findings support PAPs as first-class physical operators for reusable robot action knowledge. A PAP's identity resides in the retained response operator, instantiated by its source condition, shared model, and physical conventions. This perspective defines action objects through the physical structure they expose for prediction and control. The same operator supports **Query**, **Invoke**, **Substitute**, and **Correct** across planning, execution, monitoring, and adaptation, providing a foundation for robot systems that accumulate interaction experience as reusable physical action knowledge.

2 Related Work

Action primitives and reusable skills. Action primitives organize robot experience into reusable units of behavior. Options formalize temporal abstraction (Sutton et al., 1999), while dynamic and probabilistic movement primitives encode adaptable motions and trajectory distributions (Ijspeert et al., 2013; Paraschos et al., 2013; Saveriano et al., 2023). Learned approaches extract latent plans and skill priors (Lynch et al., 2020; Pertsch et al., 2021), infer behavioral segments and options from demonstrations (Kipf et al., 2019; Shankar & Gupta, 2020; Zhu et al., 2022), and discover primitives from offline experience (Ajay et al., 2021). Unsupervised skill discovery further uses behavioral diversity (Eysenbach et al., 2019), predictable skill dynamics (Sharma et al., 2020), controllability (Park et al., 2023), and temporal-distance geometry (Park et al., 2024). These approaches provide several foundations for reuse. We investigate local physical response as an organizing principle: do interactions considered similar exhibit compatible responses when interventions change within matched contexts? PAPs retain this intervention-response structure as an explicit action object.

Affordances, action effects, and interaction dynamics. Affordance models represent relationships among objects, actions, and effects (Montesano et al., 2008; Ardón et al., 2020), including equivalence relationships grounded in resulting effects (Andries et al., 2018). Learned dynamics models predict physical consequences for policy search and model-predictive control (Deisenroth & Rasmussen, 2011; Nagabandi et al., 2018; Chua et al., 2018). System identification and adaptation methods also use interaction history to condition control on changing dynamics (Yu et al., 2017; Kumar et al., 2021). PAPs connect these ideas at the level of the action unit: source interactions instantiate a persistent response operator with specified physical coordinates, a query domain, and invocation conventions. The operator is retained for new requests and receiving contexts. This formulation accommodates nonlinear dynamics models that satisfy the PAP object definition, connecting physical organization with an explicit object for reuse.

From action representations to operational action objects. Object–Action Complexes formalize grounded units of sensorimotor experience (Krüger et al., 2011); skill-based symbolic models connect continuous execution to abstract planning (Konidaris et al., 2018; Achterhold et al., 2023); and language-guided systems select skills or compose calls to control interfaces (Ichter et al., 2023; Liang et al., 2023). PAPs bring operational use around a retained physical response operator. **Query** exposes predicted responses and **Invoke** uses them to select and execute interventions. **Substitute** changes the execution implementation while preserving the PAP and request; **Correct** uses measurements to update the execution mapping while preserving the PAP. Requests, predictions, executed tool motions, and object responses can therefore be related through a common physical interface. Our contribution is a definition of reusable action knowledge in which **physical organization and operational use are properties of the same retained response operator**.

3 Physical Action Primitives

Physical Action Primitives (PAPs) turn interaction experience into persistent physical response operators. The key is to preserve how outcomes change with intervention, so that experience can guide actions beyond the observations from which it was acquired. Figure 2 outlines this progression: identify a response relationship, retain it as an action knowledge object, and query that object to meet new requests and adjust execution.

3.1 PAPs as Reusable Action Knowledge Objects

Similar motions and nominal outcomes can conceal different responses to changed interventions (Fig. 3). For example, two pushes can move an object equally far while inducing different rotations when tool displacement increases. Reusable action knowledge must preserve this local relationship between intervention and physical response. We define a PAP as

$$\mathcal{P} = (F_{\mathcal{P}}, \mathcal{D}_{\mathcal{P}}, \mathcal{C}_{\mathcal{P}}), \qquad \widehat{y} = F_{\mathcal{P}}(h, \xi), \quad (h, \xi) \in \mathcal{D}_{\mathcal{P}}. \tag{1}$$

Here, $F_{\mathcal{P}}$ predicts physical response $\widehat{y}$ to intervention ξ in observed context h. The domain $\mathcal{D}_{\mathcal{P}}$ declares the context and intervention scope. The conventions $\mathcal{C}_{\mathcal{P}}$ specify physical quantities, coordinate frames, units, and reference execution and measurement timing. Together, they make the

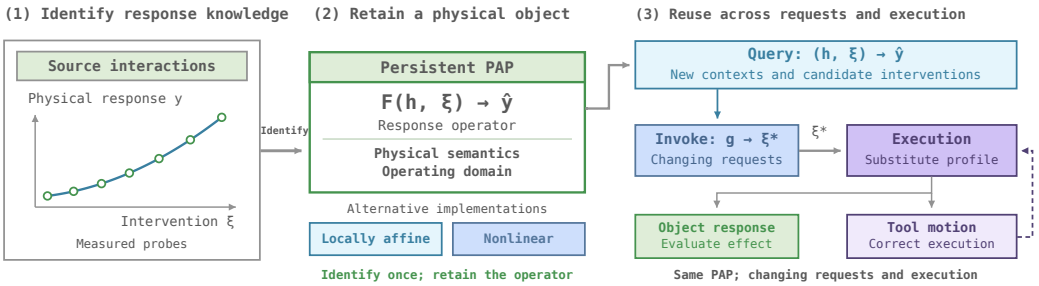


Figure 2: **From interaction experience to reusable physical action knowledge.** (1) Source probes instantiate a PAP by revealing how physical response varies with intervention. (2) The object retains a response operator, physical semantics, and an operating domain. Locally affine and nonlinear models implement this object definition. (3) Requests query and invoke the PAP. Tool-motion measurements check execution, while object-response measurements check predicted and requested effects. Execution can be substituted or calibrated while retaining the operator. The response curve in (1) illustrates the construction.

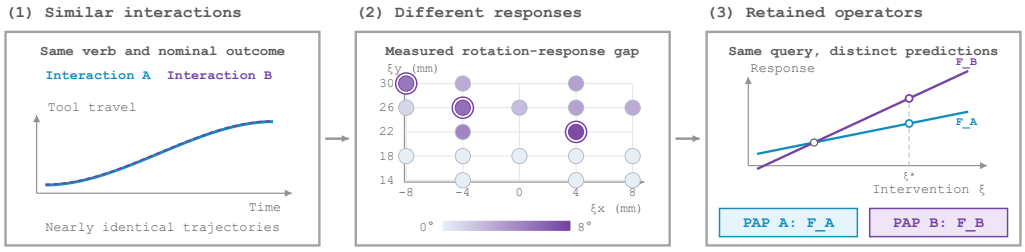


Figure 3: **Similar interactions can conceal distinct response structure.** (1) Two RollBall interactions selected under a prespecified rule have near-identical tool trajectories (maximum gap: $0.0994\,\mathrm{mm}$) and nominal displacements (gap: $0.0752\,\mathrm{mm}$). (2) Independent interventions separate their responses. Color shows the measured geodesic distance between relative object rotations in degrees; outlined markers exceed 5° (3 of 18 interventions). (3) PAPs preserve distinct response relationships as queryable operators. Curves in (1) and (3) are conceptual illustrations; points in (2) are measured data.

response relationship accessible as an action knowledge object whose predictions can be compared with measured outcomes. Appendix A specifies the physical quantities, coordinate conventions, and measurement timing.

This object provides a physical basis for organizing experience. We compare complete predicted responses over shared interventions in matched contexts, retaining both nominal effects and their variation. Independent intervention measurements then evaluate whether the resulting neighbors have compatible physical responses. The same relationship that distinguishes experiences also predicts the consequences of a proposed action, connecting physical organization to operational reuse.

3.2 Instantiating and Retaining Response Knowledge

Identify converts source probes into a retained condition $u = E(D_{\mathrm{src}})$, where D_{src} contains source trajectories, interventions, measured responses, and contexts, and E is the identification mapping. In our main construction, a frozen encoder extracts trajectory features and a learned physical projection summarizes their response evolution. Appendix B.1 describes the large-scale pretraining corpus and the 800M model supplying these features. A shared probe encoder combines these summaries with observed interventions, responses, and source contexts; averaging its outputs produces u. This aggregation retains evidence from multiple probes about how the interaction responds to intervention.

We train the resulting operator to predict measured responses to additional interventions whose trajectories and outcomes are excluded from the identification input. The model therefore learns to extend source evidence to interventions beyond its observed probes. A response-variation objective preserves changes across interventions, while step-level physical supervision grounds the projection in measured motion. Training updates the projection, probe encoder, and response model with the feature encoder frozen. Appendix B gives the objectives, normalization, architecture, and study-specific protocols.

Our main implementation exposes this learned relationship through a locally affine operator:

$$F_{\mathcal{P}}(h, \xi) = b_\theta(u, h) + A_\theta(u, h)\xi. \tag{2}$$

Here, θ denotes the response-model parameters, b_θ predicts the response at the intervention-coordinate origin, and A_θ expresses how intervention changes affect each response component. For planar manipulation, $\xi \in \mathbb{R}^2$ specifies tool displacement, and the six response components comprise planar object displacement, yaw change, terminal planar velocity, and terminal yaw angular velocity. Thus $b_\theta \in \mathbb{R}^6$ and $A_\theta \in \mathbb{R}^{6\times 2}$; both can depend nonlinearly on u and h. A source-conditioned nonlinear predictor can implement the same PAP definition through the same physical response interface.

At a new receiving state, h supplies current tool–object geometry and observed object velocities, while u supplies the response evidence retained from source interaction. This separation allows new contexts and candidate interventions to be evaluated without repeating source identification. The local scope follows the preparation conditions, intervention range, and reference execution under which the knowledge is constructed and evaluated (Appendix A.2). Retaining u, the fixed model, and these physical conventions makes the learned relationship available for subsequent requests.

3.3 Reusing PAPs across Requests and Execution

Query and Invoke. Given a retained PAP, **Query** returns $F_{\mathcal{P}}(h, \xi)$ for a proposed intervention. **Invoke** uses these predictions to select an intervention for a desired physical effect g:

$$\xi^\star = \arg \min_{\xi \in \Xi_{\mathcal{P}}(h)} \mathcal{L}_{\text{goal}}(F_{\mathcal{P}}(h, \xi), g)\,. \tag{3}$$

Here, $\Xi_{\mathcal{P}}(h)$ is the candidate intervention set for the supplied PAP and context, and $\mathcal{L}_{\text{goal}}$ measures physical discrepancy from the request. A separate execution implementation realizes the selected intervention as tool motion. When the goal changes, invocation selects a new intervention using the same retained response knowledge. Candidate grids, goal costs, and evaluation criteria are specified in Appendix D.

Substitute and Correct. Separating the response object from execution also allows the system to change how an intervention is realized. **Substitute** preserves the PAP, request, and selected displacement while replacing the execution implementation. Measurements check the resulting physical effect against the request and the retained prediction, making reuse under execution changes directly inspectable.

The interface provides two checks: requested versus measured tool motion checks intervention realization, while predicted versus measured object response checks its physical consequence. **Correct** uses the first check to calibrate the execution mapping. For the tested scale mismatch, tool-displacement measurements determine a gain correction for subsequent commands, and object responses assess the resulting effects. The same physical operator continues to guide intervention selection as execution is adjusted. Appendix E gives the temporal profiles, gain estimator, and measurement checks.

4 Experiments

Our experiments examine whether local intervention–response structure provides useful action knowledge and whether an instantiated object can continue to serve prediction, intervention selection, and execution adjustment. We first test physical compatibility under independent interventions, then examine retained source knowledge and alternative PAP constructions, and finally evaluate execution substitution and measured correction.

4.1 Experimental Setup

We evaluate PushT, PokeCube, and RollBall in ManiSkill3 (Tao et al., 2025), followed by real-robot Query and Invoke tests. The organization study uses matched RollBall contexts and models trained from scratch. The construction study evaluates frozen source-conditioned models across three simulated tasks; a separate execution study tests substitution and correction. Each study retains its original source sets, receiving states, requests, and physical criteria. Physical conventions and construction details appear in Appendices A–B; Appendices C–F specify the evaluation protocols.

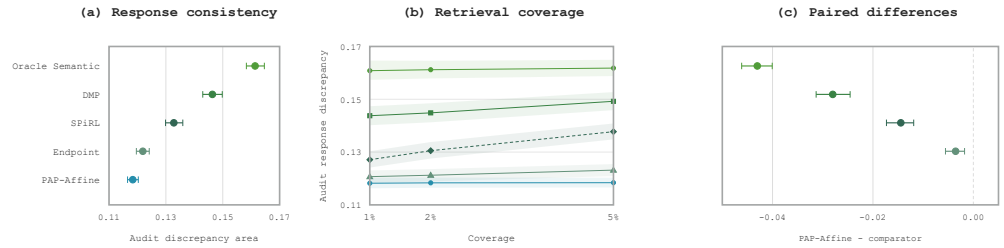


Figure 4: **Response-based organization preserves physical compatibility under independent interventions.** (a) Audit response discrepancy area with 95% intervals. (b) Mean audit discrepancy over retrieval coverage; translucent bands show 95% intervals. Method colors and top-to-bottom order follow (a); (c) retains the four comparator rows in the same order. (c) Paired PAP-Affine-minus-comparator differences with the original 99% intervals. All methods use the same gallery and lock their rankings before audit collection. Intervals resample query roots with the gallery and trained models fixed; learned-model results average three seeds. Endpoint represents a single nominal effect. The nonlinear response construction is described in Appendix C.

4.2 Response Structure Organizes Physically Compatible Experience

We ask whether interactions organized as neighbors continue to produce compatible responses when the intervention changes. We define Oracle Semantic by the true behavior label and Endpoint by the complete effect of one nominal intervention. DMP (Ijspeert et al., 2013) represents measured tool trajectories, while SPiRL (Pertsch et al., 2021) represents state–action sequences. PAP-Affine organizes the same candidate interactions through predicted responses across a shared intervention range. These comparisons test the physical information preserved by each organization criterion.

The matched RollBall study contains 320 query interactions and 640 gallery interactions from disjoint prepared contexts, with five damping conditions per context. Each response operator is identified from seven source probes. Neighbor rankings are locked before collecting responses to eighteen independent audit interventions. The audit measures displacement, relative rotation, and terminal linear and angular velocity in common physical coordinates. It therefore evaluates the organization using subsequent physical measurements. Appendix C defines the matching rules, ranking distances, audit metric, and statistical intervals.

Figure 4 shows that response-based neighborhoods have lower independently measured discrepancy than all four comparison organizations, with each paired interval below zero. PAP-Affine achieves an audit discrepancy area of 0.11834, compared with 0.16138 for Oracle Semantic and 0.14635 for DMP, corresponding to reductions of 26.7% and 19.1%, respectively. These reductions show that organizing experience by local interaction laws yields neighbors with more compatible physical responses under independent interventions. The pair in Fig. 3 makes the distinction concrete: nearly overlapping source tool motions and nominal displacements lead to distinct rotations under changed interventions. Together, the example and population audit show why preserving the response relationship adds useful physical distinctions to action knowledge.

4.3 Source Knowledge Remains Queryable and Callable

We next examine how source evidence becomes operational knowledge. Figure 5 shows three complementary observations from the recorded operator studies: prediction beyond source probes, renewed intervention selection for changing goals, and a controlled exchange of retained source information.

In the invocation example, changing the requested displacement from 6 to 9 to $12\,\mathrm{mm}$ selects tool displacements of 24.5, 27.0, and $32.0\,\mathrm{mm}$ through the same operator. Measured object displacements are 6.15, 8.55, and $13.19\,\mathrm{mm}$. The source-condition exchange tests the contribution of retained knowledge with the downstream model and control procedure fixed: mean position error changes from 0.882 to $1.166\,\mathrm{mm}$ when the paired source conditions are exchanged. These observations connect persistent identity to source-dependent physical predictions and choices. Appendix D.4 specifies the controlled source exchange and its measurement protocol.

Different constructions of the same object. The prospective three-task evaluation includes four learned PAP constructions. PAP-Affine uses an explicit locally affine response head; PAP-Nonlinear uses a nonlinear response head; PAP-Endpoint routes source information through a predicted-effect

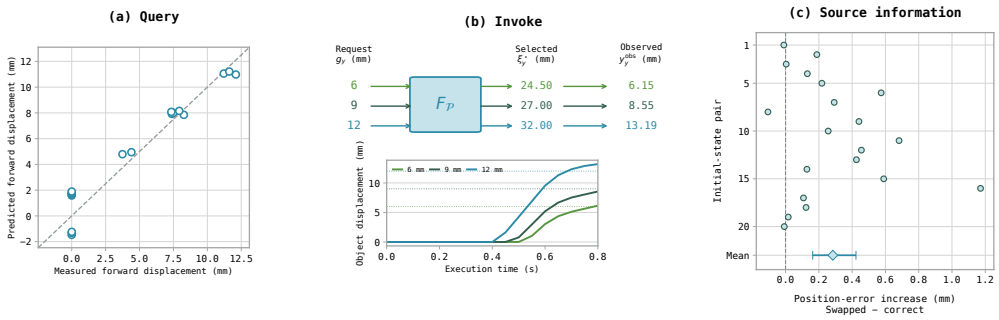


Figure 5: **Retained source knowledge supports prediction and intervention selection.** (a) Predicted and measured forward responses for eighteen non-source interventions at the first recorded RollBall parent; seven probes identify the operator. (b) One retained operator at one receiving state selects interventions for $6/9/12\,\mathrm{mm}$ requests; curves show recorded object motion and dotted lines mark requested displacements. (c) Exchanging source conditions while fixing the response model and receiving context changes position error across twenty matched initial-state pairs. The diamond and interval show the mean paired increase and its 95% bootstrap interval. Panels (a,b) are recorded operator examples; (c) is the separate matched-state damping study.

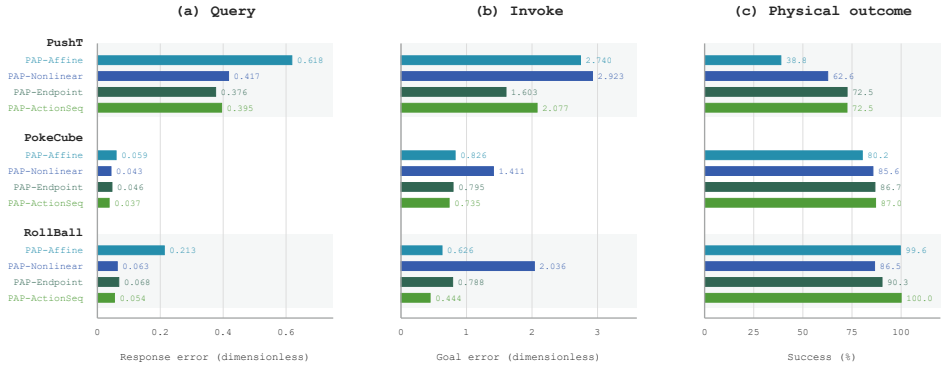


Figure 6: **Four learned PAP constructions expose a common Query–Invoke interface.** (a) Query error at new receiving states with source conditions retained. (b) Realized goal error for extrapolation requests. (c) Success under the joint position, yaw, terminal-speed, and legal-joint criteria. Errors in (a,b) are summed squared normalized residuals. Each task uses twenty source blocks, three receiving states per block, eighteen Query interventions, and nine Invoke requests. Bars average four frozen training seeds; each learned construction executes 2,160 Invoke calls per task.The source inputs, candidate interventions, selector, and execution rules are shared.

bottleneck; and PAP-ActionSeq trains its source representation with action-sequence reconstruction. Each retains a source condition and exposes a context–intervention response interface used by the same intervention selector. PAP-Endpoint here is a learned response object, distinct from the nominal-effect organizer in the preceding study. Architectures and objectives are given in Appendix B; the shared Query–Invoke protocol is detailed in Appendix D.

Figure 6 reports their predictions and measured invocation outcomes under the common interface. For each source block, the identified condition is retained across receiving states and requests, while the current context determines the response to each candidate intervention. This evaluation gives the model-class-agnostic object definition concrete implementations: different constructions provide response knowledge that can be queried and used for physical goal selection.

4.4 Retained Operators Support Execution Adjustment

The separate execution study holds the identified operator fixed while changing its downstream realization. We reuse the locked PAP-Affine model across 24 receiving states and three requests per task, yielding 72 calls per condition. Table 1 retains all three tasks and the original physical success criteria. Appendix E gives the temporal profiles, calibration equations, and reset protocol.

Substitute. Linear, smoothstep, and front-loaded profiles realize the same selected spatial intervention with different temporal progress. The recorded tool motions confirm that execution changes, while the PAP, request, and intervention remain fixed. Table 1 reports the resulting physical out-

Table 1: **Operational use under execution changes.** Entries report successful calls out of 72 per condition for PAP-Affine in the separate execution study. Substitute preserves the PAP, request, and selected spatial intervention while replacing its temporal realization. Correct preserves the PAP and request while updating the execution mapping from measured TCP motion after a $0.55\times$ scale mismatch.

	Substitute			Correct		
Environment	Linear	Smoothstep	Front-loaded	Normal	Mismatch	Corrected
PushT	23/72	22/72	22/72	23/72	0/72	1/72
PokeCube	60/72	58/72	62/72	60/72	0/72	59/72
RollBall	71/72	72/72	72/72	72/72	0/72	72/72

comes for each profile. This tests the continued use of retained response knowledge under measured changes to execution.

Correct. We inject a $0.55\times$ displacement-scale mismatch into the execution mapping. Two tool-motion measurements estimate a correction gain without reading the injected value, and subsequent calls use the calibrated mapping with the PAP unchanged. On PokeCube and RollBall, successful calls change from $0/72$ under the mismatch to $59/72$ and $72/72$ after correction, compared with reference counts of $60/72$ and $72/72$. Table 1 shows the complete task-wise outcomes.On PushT, scale calibration yields $1/72$ successful calls, compared with $23/72$ in the reference condition, and does not recover reference-level task success under the recorded paired-restoration protocol (Appendix E). Tool measurements guide the adjustment of intervention realization; object measurements determine whether the requested effect is achieved. Across these operations, the retained physical operator continues to provide the knowledge used for intervention selection.

4.5 Real-World Prediction and Invocation with PAPs

We test whether response knowledge instantiated from real interactions can support prediction and intervention selection on a Franka Emika Panda. For PushT and PokeCube, real source interactions instantiate PAPs through the frozen feature pipeline and response model. These objects expose the same state, intervention, and physical-response semantics used in simulation. Object motion is evaluated through physical measurements; position MAE averages planar Euclidean prediction errors. Figure 7 shows the PushT setup and results for both tasks. Hardware and measurement conventions are described in Appendix F.

Query: predicting physical effects. We query the instantiated PAPs to predict object responses to subsequent interventions. Across 12 trials per task, position MAE is $3.673\,\mathrm{mm}$ for PushT and $1.289\,\mathrm{mm}$ for PokeCube. These measurements assess the response knowledge against physical outcomes on hardware.

Invoke: selecting interventions for requested effects. We use the retained PAPs to select interventions for forward object displacements of 5 and $10\,\mathrm{mm}$, with 20 trials per task and request. An invocation succeeds when the Euclidean error between measured and requested planar displacement is at most $5\,\mathrm{mm}$ and the absolute wrapped error between measured and requested yaw change is at most 5°. For PushT, successful calls number $10/20$ and $12/20$, respectively; for PokeCube, they number $13/20$ and $12/20$. As the request increases from 5 to $10\,\mathrm{mm}$, mean measured displacement increases from 3.229 to $7.978\,\mathrm{mm}$ for PushT and from 3.872 to $8.313\,\mathrm{mm}$ for PokeCube. The larger requested effect therefore produces a larger measured response in both tasks through the retained PAP interface. Together, the hardware results demonstrate that real interaction evidence can become callable response knowledge for physical prediction and goal-directed intervention selection.

5 Discussion

PAPs as a foundation for robot control. PAPs suggest an architectural direction in which robot control is organized around persistent physical action knowledge. Evidence from multiple interventions can be retained as a local interaction law: a relationship between intervention and physical response that can be retained, queried, and invoked. Our experiments establish concrete capabilities

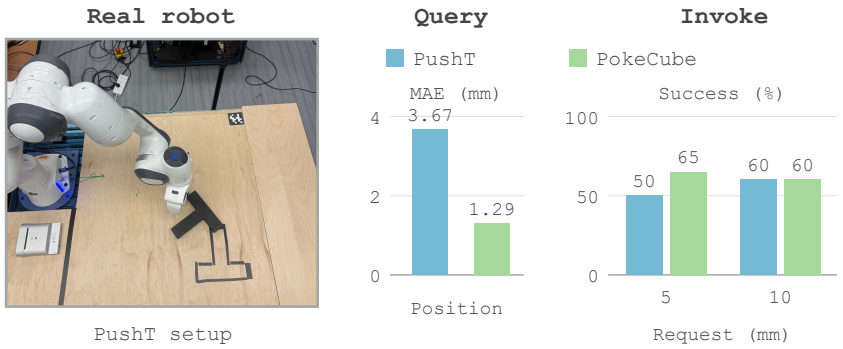


Figure 7: **Real-world prediction and invocation with PAPs.** Left: the Panda PushT setup. Center: Query position MAE over 12 trials per task. Right: Invoke success rates over 20 trials per task and requested forward displacement. Success requires both planar displacement error at most 5 mm and yaw-change error at most 5°.

of these objects: they distinguish physically different experiences, predict responses beyond source probes, and guide intervention selection while execution implementations are adjusted. Together, these capabilities motivate a physical knowledge layer between high-level reasoning and low-level control. Different response models can instantiate this layer through a common object definition, allowing the architecture to evolve while preserving explicit physical semantics.

An inspectable interface for reasoning and execution. A PAP exposes the physical consequences associated with a proposed intervention. This gives an upper-level system quantities it can reason about: desired effects, predicted responses, operating domains, and measured outcomes. An optimizer could search over interventions using PAP predictions; a planner could select operators whose predicted effects advance a task; and an LLM could translate an instruction into physical requests evaluated through the same interface. The experiments demonstrate the underlying access mechanisms: Query exposes response predictions, Invoke uses them to select interventions, and separate tool and object measurements check realization and consequence. Execution calibration further shows how a measured discrepancy can guide an implementation adjustment while the retained response object continues to support control. These interfaces make physical action knowledge available for inspection and computation throughout the control process.

Toward libraries of physical action knowledge. The broader opportunity is to accumulate robot experience as libraries of local interaction laws. Each new interaction could refine an existing operator, establish its usable domain, or supply evidence for a new object. Planning could then operate over predicted physical effects, while execution selects implementations suited to the current robot and scene. Composition would connect operators through their predicted outcomes and receiving contexts. This creates a research agenda spanning operator identification, retrieval, composition, and adaptation, with persistent physical knowledge providing a shared foundation.

The present study develops this foundation through local manipulation with explicit state and response measurements. Learning operators from these modalities and extending their use to longer-horizon tasks with changing contact conditions are important next steps toward this architecture.

6 Conclusion

We introduced Physical Action Primitives (PAPs) as reusable action knowledge objects that preserve local interaction laws: how physical responses change with intervention. Independent physical audits support these laws as a basis for organizing physically compatible experience. Affine and nonlinear constructions instantiate this knowledge as persistent operators that predict responses beyond source probes and guide intervention selection for new requests. Execution substitution and measured calibration retain the response object while adjusting how interventions are realized, and real-robot experiments demonstrate prediction and invocation on hardware. **PAPs make local interaction laws a persistent, callable unit of robot action knowledge.** This provides a foundation for robot experience to accumulate as libraries of physical operators, accessible through a common interface for prediction and control.

AI USE STATEMENT

Generative AI assisted with language polishing and figure preparation.

REPRODUCIBILITY STATEMENT

The appendices specify physical semantics and operating domains, source identification and training, evaluation metrics and protocols, execution calibration, and real-robot setup and measurement. The figures and tables report recorded experimental results under these protocols.

REFERENCES

Jan Achterhold, Markus Krimmel, and Joerg Stueckler. Learning temporally extended skills in continuous domains as symbolic actions for planning. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 225–236, 2023. URL https://proceedings.mlr.press/v205/achterhold23a.html.

Anurag Ajay, Aviral Kumar, Pulkit Agrawal, Sergey Levine, and Ofir Nachum. OPAL: Offline primitive discovery for accelerating offline reinforcement learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=V69LGwJ0lIN.

Mihai Andries, Ricardo Omar Chavez-Garcia, Raja Chatila, Alessandro Giusti, and Luca Maria Gambardella. Affordance equivalences in robotics: A formalism. *Frontiers in Neurorobotics*, 12: 26, 2018. doi: 10.3389/fnbot.2018.00026.

Paola Ardón, Èric Pairet, Katrin S. Lohan, Subramanian Ramamoorthy, and Ronald P. A. Petrick. Affordances in robotic tasks—a survey, 2020. URL https://arxiv.org/abs/2004.07400.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://papers.nips.cc/paper_files/paper/2018/hash/3de568f8597b94bda53149c7d7f5958c-Abstract.html.

Marc Peter Deisenroth and Carl Edward Rasmussen. PILCO: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on Machine Learning*, 2011. URL https://mlg.eng.cam.ac.uk/pub/pdf/DeiRas11.pdf.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SJx63jRqFm.

Brian Ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, Dmitry Kalashnikov, Sergey Levine, Yao Lu, Carolina Parada, Kanishka Rao, Pierre Sermanet, Alexander T. Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Mengyuan Yan, Noah Brown, Michael Ahn, Omar Cortes, Nicolas Sievers, Clayton Tan, Sichun Xu, Diego Reyes, Jarek Rettinghouse, Jornell Quiambao, Peter Pastor, Linda Luu, Kuang-Huei Lee, Yuheng Kuang, Sally Jesmonth, Nikhil J. Joshi, Kyle Jeffrey, Rosario Jauregui Ruano, Jasmine Hsu, Keerthana Gopalakrishnan, Byron David, Andy Zeng, and Chuyuan Kelly Fu. Do as i can, not as i say: Grounding language in robotic affordances. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 287–318, 2023. URL https://proceedings.mlr.press/v205/ichter23a.html.

Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. Dynamical movement primitives: Learning attractor models for motor behaviors. *Neural Computation*, 25 (2):328–373, 2013. doi: 10.1162/NECO_a_00393.

Thomas Kipf, Yujia Li, Hanjun Dai, Vinicius Zambaldi, Alvaro Sanchez-Gonzalez, Edward Grefenstette, Pushmeet Kohli, and Peter Battaglia. CompILE: Compositional imitation learning and

execution. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3418–3428, 2019. URL https://proceedings.mlr.press/v97/kipf19a.html.

George Konidaris, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. From skills to symbols: Learning symbolic representations for abstract high-level planning. *Journal of Artificial Intelligence Research*, 61:215–289, 2018. doi: 10.1613/jair.5575.

Norbert Krüger, Christopher Geib, Justus Piater, Ronald Petrick, Mark Steedman, Florentin Wörgötter, Aleš Ude, Tamim Asfour, Dirk Kraft, Damir Omrčen, Alejandro Agostini, and Rüdiger Dillmann. Object–action complexes: Grounded abstractions of sensory–motor processes. *Robotics and Autonomous Systems*, 59(10):740–757, 2011. doi: 10.1016/j.robot.2011.05.009.

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: Rapid motor adaptation for legged robots. In *Proceedings of Robotics: Science and Systems*, 2021. doi: 10.15607/RSS.2021.XVII.011.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *IEEE International Conference on Robotics and Automation*, pp. 9493–9500, 2023. doi: 10.1109/ICRA48891.2023.10160591. URL https://arxiv.org/abs/2209.07753.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

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

Luis Montesano, Manuel Lopes, Alexandre Bernardino, and José Santos-Victor. Learning object affordances: From sensory–motor coordination to imitation. *IEEE Transactions on Robotics*, 24 (1):15–26, 2008. doi: 10.1109/TRO.2007.914848.

Anusha Nagabandi, Gregory Kahn, Ronald S. Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *IEEE International Conference on Robotics and Automation*, 2018. URL https://arxiv.org/abs/1708.02596.

Alexandros Paraschos, Christian Daniel, Jan R. Peters, and Gerhard Neumann. Probabilistic movement primitives. In *Advances in Neural Information Processing Systems*, volume 26, 2013. URL https://papers.nips.cc/paper/2013/hash/e53a0a2978c28872a4505bdb51db06dc-Abstract.html.

Seohong Park, Kimin Lee, Youngwoon Lee, and Pieter Abbeel. Controllability-aware unsupervised skill discovery. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 27225–27245, 2023. URL https://proceedings.mlr.press/v202/park23h.html.

Seohong Park, Oleh Rybkin, and Sergey Levine. METRA: Scalable unsupervised RL with metric-aware abstraction. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=YgZNmDqyR6.

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

Matteo Saveriano, Fares J. Abu-Dakka, Aljaž Kramberger, and Luka Peternel. Dynamic movement primitives in robotics: A tutorial survey. *The International Journal of Robotics Research*, 42(13):1133–1184, 2023. doi: 10.1177/02783649231201196.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Tanmay Shankar and Abhinav Gupta. Learning robot skills with temporal variational inference. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8624–8633, 2020. URL https://proceedings.mlr.press/v119/shankar20b.html.

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJgLZR4KvH.

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

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-Kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh Nagaswamy Rajesh, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Demonstrating GPU parallelized robot simulation and rendering for generalizable embodied AI with ManiSkill3. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.021. URL https://www.roboticsproceedings.org/rss21/p021.html.

Wenhao Yu, Jie Tan, C. Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. In *Proceedings of Robotics: Science and Systems*, 2017. doi: 10.15607/RSS.2017.XIII.048.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1703.06114.

Yifeng Zhu, Peter Stone, and Yuke Zhu. Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation. *IEEE Robotics and Automation Letters*, 7(2):4126–4133, 2022. doi: 10.1109/LRA.2022.3146589.

A Physical Semantics and Operating Domains

A.1 Interventions, responses, and retained information

An instantiated PAP consists of a response operator, a domain, and physical conventions. In the learned constructions, its stored state comprises the source condition u, a reference to the fixed identification and response-model parameters, the normalization constants, and the coordinate and timing conventions. The source condition is computed once from the source packet. A call supplies the current context h and a candidate intervention ξ; it does not update u or the response-model weights.

In the planar experiments, $\xi = (\xi_x, \xi_y)$ is a requested tool-center-point (TCP) displacement in metres. The local y direction is the prescribed forward interaction heading, and x is the perpendicular direction in the table plane. Let R_ϕ rotate this local frame into the world frame, where ϕ is the heading fixed for the interaction. For object position p_t, yaw ψ_t, linear velocity v_t, and angular velocity ω_t, the measured response is

$$y = \left[\left(R_\phi^\top (p_T - p_0)\right)_{xy},\ \mathrm{wrap}(\psi_T - \psi_0),\ \left(R_\phi^\top v_T\right)_{xy},\ \omega_{T,z} \right] \in \mathbb{R}^6. \tag{4}$$

Here, 0 denotes the state immediately before the intervention and T the end of the complete execution horizon, including the hold period. The function $\mathrm{wrap}(a) = \mathrm{atan2}(\sin a, \cos a)$ expresses yaw differences in $[-\pi, \pi]$. Terminal velocities are measured velocities, rather than differences from their initial values.

Table 2: Planar preparation and intervention parameters. Lengths are in millimetres. The lateral preparation offset alternates between zero and the listed value. The same r applies to both intervention axes.

Task	c_y	r	Object back	Tool extent	Height	Offset
PushT	26	8	62.5	8	24	30
PokeCube	15	6	20	18	25	10
RollBall	22	8	35	18	35	12

The planar context has 12 components: three TCP–object relative-position components, three observed object linear velocities, three angular velocities, and three public geometry quantities specifying object back extent, tool extent, and preparation height. Relative planar position is heading-aligned; linear and angular velocity inputs follow the recorded world-state convention. Before standardization, lengths are divided by $0.1\,\mathrm{m}$ and linear velocities by $0.05\,\mathrm{m/s}$; angular velocities are represented in $\mathrm{rad/s}$. Each component is then standardized with its training mean and a training standard deviation floored at 0.01. The retained u supplies source-response evidence, while h supplies the state and geometry for the current query. Hidden damping values and task identity labels are not additional context inputs.

The physical-organization study uses a 12-component rigid-body response. It contains local three-dimensional displacement, a local relative-rotation vector, and terminal local linear and angular velocities. If $Q_0, Q_T \in \mathrm{SO}(3)$ are initial and final orientations, the rotation vector is obtained from $\log(Q_T Q_0^{-1})$ and rotated into the heading-aligned frame. Rotation comparisons use geodesic distance between the corresponding rotations, as defined in Appendix C. The context and trajectory encoder in that study instead receive the recorded object state, TCP pose, and robot joint positions and velocities.

A.2 Physical normalization and query ranges

Planar networks receive $\bar{\xi} = \xi/(0.01\,\mathrm{m})$ and predict normalized responses $z = S_6^{-1} y$, with diagonal scale matrix

$$S_6 = \mathrm{diag}\,(0.005\,\mathrm{m}, 0.005\,\mathrm{m}, \pi/36\,\mathrm{rad}, 0.05\,\mathrm{m/s}, 0.05\,\mathrm{m/s}, 1\,\mathrm{rad/s})\,. \tag{5}$$

The rigid-body construction repeats the corresponding displacement, rotation, linear-velocity, and angular-velocity scales over each group's three components. Its intervention coordinates are centered at $(0, 22)\,\mathrm{mm}$ and divided by $8\,\mathrm{mm}$. These transformations are absorbed into the coefficients when the operator is written in physical units in the main text.

For planar source collection, define the normalized grid $G = \{-1, -0.5, 0, 0.5, 1\}^2$. A grid coordinate $a \in G$ maps to $\xi = c + ra$, where $c = (0, c_y)$ is the nominal forward displacement and r is a task-specific half-width. Seven coordinates identify the object:

$$G_{\mathrm{src}} = \{(0,0), (-1,0), (1,0), (0,-1), (0,1), (-1,-1), (1,1)\}. \tag{6}$$

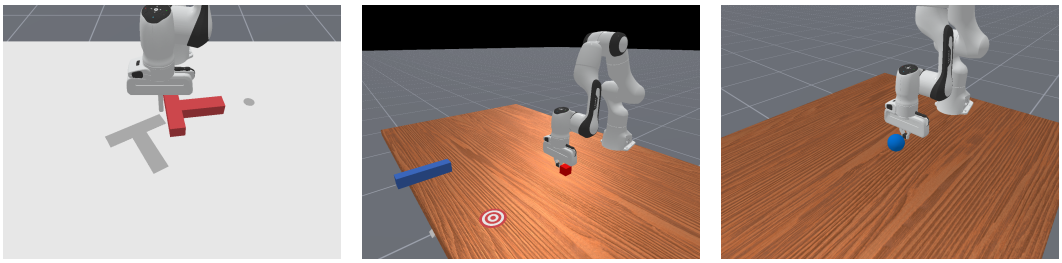
The remaining 18 coordinates supply prediction targets or evaluation queries. In the implementation's lexicographic grid order, the source indices are $[12, 2, 22, 10, 14, 0, 24]$. The nominal grid point is a nonzero physical displacement.

The standard invocation grid uses nine evenly spaced coordinates per axis in $[-1.25, 1.25]$; the expanded grid uses $[-2.5, 2.5]$. Both are mapped by $c + ra$, giving 81 candidates. Source bounds, candidate bounds, and the measured evaluation population specify different parts of the operating domain. The domains used for reported outcomes are identified with their protocols in Appendices D and E.

A.3 Preparation, control, and timing

Figure 8 shows the three simulation environments used for planar PAP construction and evaluation.

Planar collection samples object-center offsets within $\pm 18\,\mathrm{mm}$ along both table axes. PushT and PokeCube sample yaw within $\pm 0.15\,\mathrm{rad}$; RollBall retains its prescribed heading of π. The robot first raises the TCP, moves behind the object, and lowers it to the task-specific height. Preparation runs



(a) PushT (b) PokeCube (c) RollBall

Figure 8: **Simulation environments for learning and using PAPs.** PushT, PokeCube, and RollBall provide distinct object geometries and contact responses for source-probe collection and physical invocation. The views render saved robot and object states at the start of recorded calls. Native scene target markers belong to the simulator task; PAP requests specify local physical displacements.

for 180 control steps. Stage transitions use TCP height above $0.13\,\mathrm{m}$, planar approach error below $3\,\mathrm{mm}$, and final height error below $4\,\mathrm{mm}$. The behind-object reference includes a $9\,\mathrm{mm}$ clearance beyond the object and tool extents in Table 2.

The simulated planar experiments use a Cartesian displacement controller. Desired world-frame TCP error is rotated into the robot base frame, divided by $0.1\,\mathrm{m}$, and clipped componentwise to $[-0.2, 0.2]$ before being passed to the controller. The gripper is held closed where present. PushT uses the simulated pushing tool; PokeCube uses direct closed-gripper contact with the unused peg parked away from the interaction. The reference planar execution has 12 movement steps followed by four hold steps at $20\,\mathrm{Hz}$. The organization study uses 16 movement and eight hold steps; the source-condition exchange uses eight movement and eight hold steps. Each response is evaluated at its own protocol's final step. Timing is part of the physical convention of that experiment.

B Learning and Instantiating PAPs

B.1 Large-scale interaction data and the pretrained backbone

The planar PAP construction builds on a frozen encoder from an approximately 800-million-parameter pretrained model. Its interaction corpus combines newly collected manipulation experience, historical exploration, and controlled branches from shared initial states. The fresh collection contains 7,962,624 episode roots and 539,688,960 state transitions. Historical exploration adds 1,945,600 transitions, and 423,936 eight-step branches from 52,992 parent states add 3,391,488 transitions. The combined corpus contains 545,026,048 transitions across training and development partitions. Fresh source-training tasks are PokeCube, PullCube, PushT, and RollBall; StackCube supplies the development partition. The historical supplement adds five manipulation tasks. PPO policies (Schulman et al., 2017) with controller-noise levels of 0, 0.05, 0.15, and 0.30 produce the fresh trajectories, preserving both successful and unsuccessful interactions.

The full pretrained model has 796,957,496 parameters and learns from action, physical-effect, and state-transition reconstruction objectives. Its encoder has width 2,032 and 24 residual MLP blocks, each with feedforward multiplier four. Training uses 100,000 optimizer updates on four NVIDIA B200 GPUs, with an effective batch of 128 trajectory windows and BF16 computation. AdamW uses learning rate 10^{-4}, 2,000 warmup updates, cosine decay to one tenth of the initial rate, and weight decay 0.01. The run processes 12.8 million window presentations; corpus size and optimizer exposure describe the data resource and the training budget, respectively. PAP construction then reuses the frozen encoder features and trains the physical response interface on the source and target probes specified below.

Backbone, physical translator, and executor. These components have distinct roles in the implemented system. The backbone encodes observed interaction histories. The physical translator comprises the step-level projection and probe aggregation that form u, together with the response head that maps (u, h, ξ) to physical predictions. For the main affine construction, these trainable modules contain 326,329 parameters in total, including the auxiliary readiness head. Invoke uses the retained response model to choose a tool displacement; a separate Cartesian controller realizes that displacement through the prescribed motion profile. Execution calibration updates the displacement mapping from tool measurements while preserving the backbone and retained response model. This

Table 3: Four constructions of the planar PAP interface. Each retains a 32-dimensional source condition and accepts current context and candidate intervention. Head descriptions refer to normalized response coordinates.

Construction	Response parameterization	Construction-specific supervision
PAP-Affine	$\bar{b}(u, \widetilde{h}) + \bar{A}(u, \widetilde{h})\bar{\xi}$	Physical step projection and response objectives
PAP-Nonlinear	$f(u, \widetilde{h}, \bar{\xi})$	Same physical step and response objectives
PAP-Endpoint	$f(e(u), \widetilde{h}, \bar{\xi})$, with $e(u) \in \mathbb{R}^6$	Additional nominal-effect prediction loss on $e(u)$
PAP-ActionSeq	$f(u, \widetilde{h}, \bar{\xi})$	Source action-sequence reconstruction replaces physical step supervision

division allows a shared large-scale representation to support compact physical knowledge objects and independently adjustable execution.

B.2 Source packets and frozen feature extraction

For each planar source condition, the seven probes in Eq. 6 provide trajectories, requested interventions, measured responses, and the source context. Each trajectory has 16 control steps. The frozen feature encoder uses the canonical input, a projection to width 2,032, 24 residual MLP blocks with a feedforward multiplier of four, and a final layer normalization. Feature extraction processes two consecutive eight-step windows and produces a 2,032-dimensional vector at each step. We use the final normalized encoder features before the representation's compression heads.

The encoder inputs follow the checkpoint's canonical normalization and concatenate state, action, state change, physical effect, and contact-availability information. Controller actions are encoded from the realized arm joint targets: the target minus the pre-step joint position, divided by $0.1\,\mathrm{rad}$, plus the recorded gripper command where applicable. Missing contact-force channels are zero-filled with their availability masks set to false. Requested Cartesian commands are not substituted for measured joint-target inputs. Encoder features are cached in half precision; the encoder receives no updates during PAP construction.

Let e_{it} be the frozen feature at step t of probe i. A learned projection P produces six normalized physical quantities $\widetilde{s}_{it} = P(e_{it})$. Its first three components estimate step displacement and yaw change; its last three estimate planar velocity and yaw angular velocity. The temporal summary, probe encoding, and retained condition are

$$\begin{aligned} v_i &= \left[\sum_{t=1}^{T} \widetilde{s}_{it,1:3},\ \widetilde{s}_{iT,4:6} \right], \\ u &= \frac{1}{7} \sum_{i=1}^{7} E_{\mathrm{probe}} \left([\bar{\xi}_i, z_i, v_i, \widetilde{h}_i] \right). \end{aligned} \tag{7}$$

Here $\widetilde{h}_i$ is the standardized source context and z_i is the normalized measured response. Both P and E_{probe} are MLPs with two hidden layers of width 128 and SiLU activations; the probe encoder outputs 32 components. Mean pooling gives the 32-dimensional u, using permutation-invariant aggregation (Zaheer et al., 2017). No discrete primitive labels are assigned during identification.

B.3 Response heads and alternative constructions

Table 3 summarizes the four PAP implementations. The main response head receives $[u, \widetilde{h}]$ and outputs six bias terms, twelve intervention coefficients, and one preparation-readiness logit. Its MLP has two width-128 hidden layers with SiLU activations. In normalized coordinates, $\widehat{z} = \bar{b}(u, \widetilde{h}) + \bar{A}(u, \widetilde{h})\bar{\xi}$, where $\bar{A} \in \mathbb{R}^{6\times 2}$. The physical operator returns $S_6\widehat{z}$.

All added networks use the same two-hidden-layer, width-128 MLP template. PAP-Endpoint compresses u through a six-component nominal-effect bottleneck but retains dependence on candidate

intervention and current context. PAP-ActionSeq reconstructs each source probe's complete 16×8 controller-action sequence from its retained condition and source intervention. Its output remains a physical response prediction, used by the common intervention selector. The nominal-effect organizer called Endpoint in Appendix C is a different object: it represents one measured effect directly.

B.4 Learning from source probes to unobserved interventions

Let J index all 25 training interventions, let $I_{\mathrm{src}} \subset J$ index the seven source coordinates in Eq. 6, and let $H = J \setminus I_{\mathrm{src}}$ denote the 18 target indices absent from the identification packet. Let $\ell(a, b)$ be Smooth L1 loss, averaged over all indicated components and examples, with its standard unit transition between quadratic and linear penalties. The response and variation losses are

$$\begin{aligned} \mathcal{L}_{\mathrm{pred}} &= \frac{1}{|H|} \sum_{j \in H} \ell(\widehat{z}_j, z_j), \\ \mathcal{L}_{\mathrm{var}} &= \frac{1}{25} \sum_{j \in J} \ell(\widehat{z}_j - \overline{\widehat{z}}, z_j - \bar{z}), \end{aligned} \tag{8}$$

where bars denote means over all 25 interventions for the same condition. The first term trains prediction beyond the source probes; the second constrains intervention-dependent changes. The additional 18 trajectories and responses enter supervision, not the computation of u.

The planar objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{pred}} + 0.5\mathcal{L}_{\mathrm{var}} + \mathcal{L}_{\mathrm{step}} + 0.2\mathcal{L}_{\mathrm{match}} + 0.2\mathcal{L}_{\mathrm{ready}} + 0.5\mathcal{L}_{\mathrm{pref}}. \tag{9}$$

The terms have the following definitions and responsibilities.

Step supervision. $\mathcal{L}_{\mathrm{step}}$ compares $\widetilde{s}_{it}$ with measured normalized step displacement, wrapped yaw increment, post-step planar velocity, and yaw angular velocity. It trains the physical projection directly and influences the probe summary through Eq. 7.

Matched-response consistency. For each member of a training minibatch, the nearest other standardized context is considered. A pair is accepted when its context distance is below 0.5 and the largest absolute difference across all seven normalized source responses and all six components is below one. $\mathcal{L}_{\mathrm{match}}$ is Smooth L1 loss between the two predicted 25-intervention response arrays, averaged over accepted pairs; it is zero when no pair qualifies.

Preparation readiness. $\mathcal{L}_{\mathrm{ready}}$ is binary cross-entropy between the readiness logit and the indicator that preparation reached its final stage. This auxiliary head trains the shared source representation. Query and Invoke use the response outputs; the readiness logit is not a separately calibrated probability of invocation success.

Intervention preference. One measured training response z_k is sampled as the normalized goal for each example. Define predicted and measured costs $c_j = \|\widehat{z}_j - z_k\|_2^2/6$ and $c_j^{\mathrm{obs}} = \|z_j - z_k\|_2^2/6$. The preference loss is cross-entropy between $\mathrm{softmax}(-4c^{\mathrm{obs}})$ and $\mathrm{softmax}(-4c)$, with the measured distribution treated as a fixed target. This connects response learning to relative intervention selection.

PAP-Endpoint adds $\ell(e(u), z_{\mathrm{nominal}})$ with weight one. PAP-ActionSeq replaces $\mathcal{L}_{\mathrm{step}}$ with a weight-one Smooth L1 action-sequence reconstruction loss; the remaining response objectives are retained. The projection, probe encoder, and response head receive gradients through the active objectives. The frozen feature encoder does not.

B.5 Training configuration and model selection

Planar training uses 300 prepared conditions per task and development uses 60 per task, giving 900 training and 180 development conditions pooled across PushT, PokeCube, and RollBall. Each condition provides a 25-intervention panel. Each construction is trained for 10,000 updates with AdamW (Loshchilov & Hutter, 2019), learning rate 3×10^{-4}, weight decay 10^{-5}, minibatch size 32, and

gradient-norm clipping at one. Four seeds, 104000–104003, are used for each construction. Every 500 updates, development Smooth L1 error on the 18 non-source interventions selects a checkpoint within that seed.

The three-task prospective evaluation retains all four selected checkpoints per construction and averages their outcomes. The recorded single-operator and execution studies use the development-selected affine checkpoint (seed 104002). Selection precedes their evaluation outcomes. Model parameters, feature normalization, and source identification remain fixed during evaluation.

The organization study has its own construction. A per-step MLP maps raw object state, TCP pose, joint position, and joint velocity to 32 features, which are averaged over time. A second MLP combines this summary with normalized source intervention, the 12-component source response, and standardized initial state; mean pooling over seven probes produces u. An affine rigid-response head or nonlinear MLP predicts the 12-component response. Training uses Smooth L1 loss on the 18 non-source interventions, with 128 training roots and 32 development roots, each at five damping settings. These correspond to 640 and 160 source conditions.

All learned organizers are trained from scratch on these training roots. AdamW uses learning rate 3×10^{-4}, weight decay 10^{-5}, batch size 64, gradient clipping at one, and 4,000 updates. Widths 128 and 256 are evaluated every 250 updates; the development objective selects width and checkpoint separately for each method and seed. The three seeds are 24140061–24140063. Source-state standard deviations are floored at 0.01; the SPiRL action standard deviations are floored at 0.001. This study uses its own source encoder rather than the frozen planar feature encoder.

C Evaluating Physical Organization

C.1 Matched interactions and independent physical evidence

The organization experiment uses RollBall with a common closed-gripper push behavior and five settings of both linear and angular damping: $0, 2, 4, 6, 8$ in the simulator's damping convention. A root denotes a prepared initial condition; its five damping variants remain grouped during statistical analysis. Damping is set to four during preparation and is changed for the interaction. Models receive recorded interaction evidence, not the damping label.

Training, development, final query, and gallery roots are disjoint. Final evaluation uses 64 query roots and 128 gallery roots, producing 320 query and 640 gallery interactions. Each interaction supplies the seven source interventions in Eq. 6, mapped to the RollBall range in Table 2. Rankings are computed from source information and locked before final audit outcomes are collected. The 18 remaining points of the 5×5 grid form the independent audit panel.

Eligible query–gallery pairs must satisfy TCP–object relative-position difference at most $2\,\mathrm{mm}$, maximum arm-joint difference at most $0.03\,\mathrm{rad}$, and maximum gripper-position difference at most $0.1\,\mathrm{mm}$. Their measured source TCP profiles must agree within $0.5\,\mathrm{mm}$ at all compared samples. Candidate sets include all five variants of each eligible gallery root. Each query must have at least 200 eligible gallery interactions. Final audit profiles are checked against the same cross-root tolerance.

Every branch is generated in a fresh simulator process using the same complete preparation recipe. Control and simulation frequencies are 20 and $100\,\mathrm{Hz}$. Source validity checks require commanded-profile error at most $3.5\,\mathrm{mm}$, endpoint error at most $0.25\,\mathrm{mm}$, and tool-orientation error at most 2°. Within-root source profiles must agree within $0.25\,\mathrm{mm}$. A source root is retained as a complete five-condition bundle. The final audit requires complete valid panels for the locked cohort, preserving the correspondence between source rankings and measured evidence.

C.2 Organization criteria and information access

Oracle Semantic. The true behavior identity is supplied. All candidate interactions in this study share that identity, so their semantic distances are tied. A fixed metadata hash orders ties independently of physical outcomes. This evaluates which physical distinctions remain within a common semantic action.

DMP. Each of the seven measured three-dimensional TCP trajectories is fitted by a discrete dynamical movement primitive (Ijspeert et al., 2013) with 12 basis functions per dimension. The representation concatenates fitted weights, initial and final TCP positions, and duration across probes. Distances are Euclidean after division by training-feature standard deviations, floored at 10^{-6}.

SPiRL. We use the official closed-loop SPiRL skill model (Pertsch et al., 2021) with a state-conditioned posterior and decoder. The implementation uses a ten-dimensional skill latent, five processing layers, and KL weight 5×10^{-4}. Training includes state–action sequences from the shared training interaction panels. At final evaluation, the posterior means from the seven source probes are concatenated. Euclidean distance uses training-feature standard deviations floored at 10^{-3}. Width and checkpoint are selected by the model's development objective, using the schedule in Appendix B.

Endpoint. The representation is the measured 12-component response to the nominal source intervention $(0, 22)\,\mathrm{mm}$. Distance uses the physical group metric below. This criterion retains one nominal effect; the seven-probe criteria retain a broader source packet. The comparison evaluates these complete organization criteria with their declared inputs.

PAP response structure. The retained operator predicts responses on a common 7×7 query grid spanning $[-8, 8]\,\mathrm{mm}$ laterally and $[14, 30]\,\mathrm{mm}$ forward. Each object is queried at its recorded context, and comparisons are restricted to the matched candidate set. Both the affine and nonlinear response constructions use this interface; the main comparison figure presents PAP-Affine against the four organization criteria above.

C.3 Response distance, coverage, and uncertainty

For two rigid responses y, y', write d_p, d_v, d_ω for Euclidean distances between their displacement, linear-velocity, and angular-velocity groups. Let r, r' be their relative-rotation vectors. Rotation distance is

$$\begin{aligned} d_R(r, r') &= \left\| \log\left(\exp([r]_\times) \exp([r']_\times)^{-1} \right) \right\|_2, \\ d_{\mathrm{phys}}(y, y') &= \frac{1}{4} \left(\frac{d_p}{0.005} + \frac{d_R}{\pi/36} + \frac{d_v}{0.05} + \frac{d_\omega}{1} \right). \end{aligned} \tag{10}$$

The group distances are expressed in metres, radians, metres per second, and radians per second, respectively; $[r]_\times$ is the skew-symmetric matrix representing cross product with r. For PAPs $\mathcal{P}, \mathcal{Q}$, the ranking distance is

$$d_{\mathrm{pred}}(\mathcal{P}, \mathcal{Q}) = \frac{1}{49} \sum_{k=1}^{49} d_{\mathrm{phys}} \left(F_{\mathcal{P}}(h_{\mathcal{P}}, \xi_k), F_{\mathcal{Q}}(h_{\mathcal{Q}}, \xi_k) \right). \tag{11}$$

This compares complete response relationships, retaining nominal response differences as well as intervention-dependent variation. The independent audit replaces the predicted responses in this average with measured responses on the 18 audit interventions. Predicted and measured arrays therefore play separate roles: the former choose neighbors and the latter evaluate them.

At coverage $c \in \{0.01, 0.02, 0.05\}$, a query with n eligible candidates retrieves $\lceil cn \rceil$ neighbors. We average audit discrepancy over those neighbors, then over the five query conditions within each root, and then equally over roots. In the final cohort, the three coverages retrieve 7, 13, and 32 neighbors per query. The area statistic is trapezoidal integration over these three coverage points, divided by $0.05 - 0.01$.

Learned-method curves are averaged over three model seeds before uncertainty estimation. We use 5,000 bootstrap resamples of the 64 query roots, carrying all five conditions together and retaining the fixed gallery and trained models. Individual curves and areas use percentile 95% intervals. Paired PAP-minus-comparator differences use the same root resamples and the original 99% intervals from the five-contrast analysis. The four displayed organization contrasts retain those intervals unchanged. Under the same response-based ranking protocol, the nonlinear and affine constructions have discrepancy areas 0.11832 and 0.11834, respectively; both instantiate the physical response criterion.

C.4 Response-structure illustration

The illustration uses the first metadata-ordered qualifying within-root pair under the recorded selection rule. A pair qualifies when its source DMP distance lies in the bottom quartile and at least two audit interventions exceed either $5\,\mathrm{mm}$ displacement difference or 5° rotation difference. The selected pair has maximum source TCP separation $0.0994\,\mathrm{mm}$ and nominal object-displacement separation $0.0752\,\mathrm{mm}$. Three of its 18 audit interventions exceed the rotation threshold. The response map displays measured geodesic rotation differences. The accompanying trajectory and retained-operator curves serve as conceptual illustrations, as specified in the figure caption.

D Query and Invoke Evaluation

D.1 Source blocks and receiving contexts

The three-task construction evaluation uses 20 independently sampled source blocks per task, each with one source state and three receiving states. Source positions are sampled within the $\pm 18\,\mathrm{mm}$ preparation range. Receiving positions perturb their source position by up to $2\,\mathrm{mm}$ per axis, and receiving yaw adds up to $0.02\,\mathrm{rad}$ to source yaw. RollBall keeps its fixed heading. Each source block retains the same alternating lateral preparation offset across its source and receivers.

Seven source probes instantiate each model's u. The primary evaluation retains that condition across all three receivers, supplying each receiver's current h. Query predicts the 18 non-source grid responses at each receiver. The response targets for those queries are excluded from identification. Evaluation executes each physical intervention in a fresh process with the same complete 80-lane population: 20 source states and 60 receivers. The lane population and preparation recipe stay fixed across interventions, and initial object, TCP, joint-position, and joint-velocity arrays are checked to agree within 10^{-5} in their recorded units.

All four training seeds per construction are fixed before this evaluation. Source identification, model predictions, candidate selections, and their records are finalized before measured invocation outcomes are used for scoring. This provides $20 \times 3 \times 18 = 1{,}080$ receiving-state Query comparisons per model seed and task.

D.2 Candidate selection and physical criteria

Define the request as $g = (g_x, g_y, 0, 0, 0, 0)$, so the desired effect specifies planar displacement, zero yaw change, and zero terminal velocities. The local goal family uses $g_x \in \{-3, 0, 3\}\,\mathrm{mm}$ and $g_y \in \{6, 9, 12\}\,\mathrm{mm}$; the expanded goal family uses the same lateral values with $g_y \in \{15, 18, 21\}\,\mathrm{mm}$. Each family has nine requests and uses its corresponding 81-candidate grid from Appendix A.2. The main four-construction figure reports fixed-source Query and expanded-goal Invoke.

Given a PAP, the selector evaluates all candidates and chooses

$$\xi^\star = \arg\min_{\xi \in \Xi(h)} \sum_{r=1}^{6} \left(\frac{F_{\mathcal{P}}(h, \xi)_r - g_r}{(S_6)_{rr}} \right)^2 . \tag{12}$$

The first minimum in the shared candidate order resolves ties. The current PAP is supplied to this optimization; no library-level PAP retrieval is performed by this selector. Each selected intervention is executed for 12 movement and four hold steps using the shared controller.

Query error is the sum of squared normalized prediction residuals. Realized goal error applies the same sum to measured response minus requested response. Both are dimensionless and average equally across their respective evaluated queries or calls. For physical interpretation, planar position error is the Euclidean norm of the two displacement residuals. Invocation success requires all of the following: position error at most $5\,\mathrm{mm}$, absolute wrapped yaw error at most 5°, terminal planar speed at most $0.05\,\mathrm{m/s}$, and every commanded arm joint target within its limits. The joint-limit comparison includes a $10^{-5}\,\mathrm{rad}$ numerical tolerance. Terminal yaw angular velocity contributes to the continuous goal cost; it has no separate threshold in this success indicator.

Each model seed executes $20 \times 3 \times 9 = 540$ calls per task and goal family. Four seeds yield 2,160 calls per construction and task in the main figure. All receiving states and unsuccessful calls con-

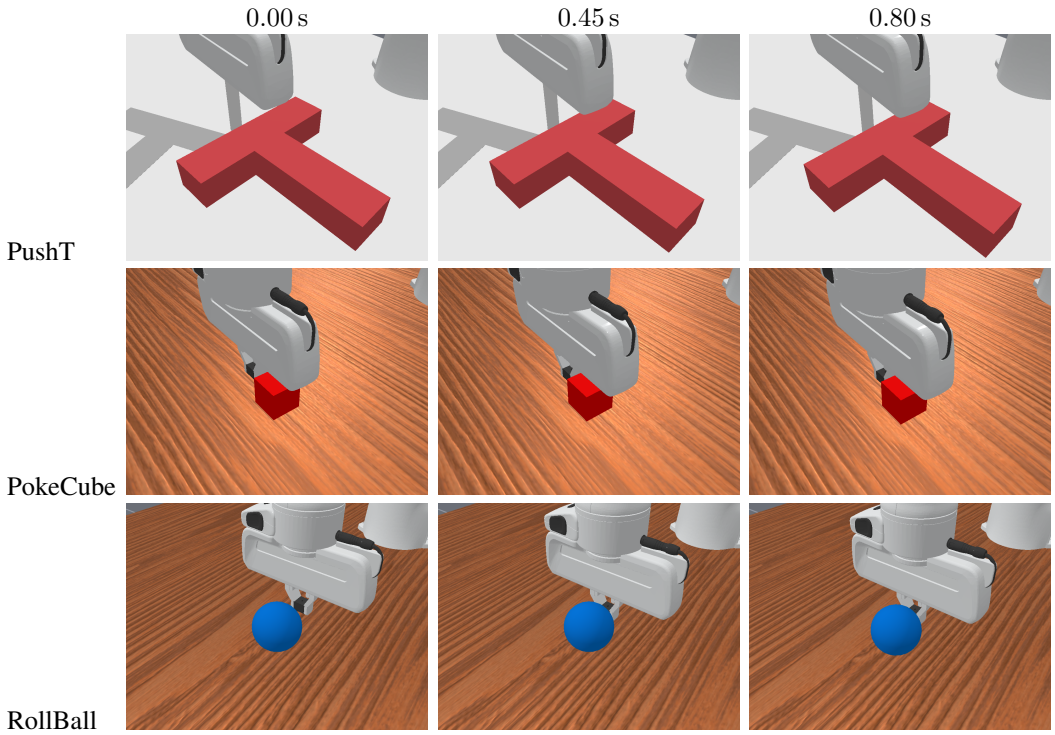


Figure 9: **PAP invocation across three simulated manipulation tasks.** Each row shows the start, an intermediate state, and the end of a recorded call requesting 9 mm forward object displacement. Frames are rendered from saved simulator states at the indicated times. The atlas selects one parent per task by the highest number of successful calls over its nine-request panel, breaking ties by mean position error; we show the central request from each selected panel.

tribute to the denominator. The plotted means average the four seed results. These results describe four implementations of the shared PAP interface.

D.3 Recorded operator examples

The Query panel of the retained-operator figure uses the first recorded RollBall parent and its 18 non-source interventions. Its seven source probes instantiate the displayed affine operator. The Invoke panel retains one operator and one receiving state while changing the requested forward displacement from 6 to 9 to 12 mm. The selected tool displacements are 24.5, 27.0, and 32.0 mm; measured object displacements are 6.15, 8.55, and 13.19 mm. These are recorded examples, rather than averages over the prospective population above. Their curves show measured object trajectories, with requested displacements drawn separately.

Visualizing physical invocation. Figure 9 shows recorded calls for a common 9 mm forward request across the three object geometries. The image sequences connect the physical request to contact, tool motion, and the resulting object state.

D.4 Controlled exchange of retained source conditions

The source-exchange experiment uses 20 matched initial-state pairs of RollBall. Each pair has the same sphere geometry and paired initial-state recipe, with linear and angular damping set to either zero or eight. The damping settings are excluded from model input. Seven probes per condition identify u; additional measured branches are not passed to identification or intervention selection. The frozen affine response model and the receiving context are held fixed when the two source conditions within a pair are exchanged.

Each condition receives forward requests of 6, 9, and 12 mm, giving 120 calls per identity assignment. Selection uses the local 81-candidate grid and the same normalized response cost. Each call has eight movement and eight hold steps. Its recorded success criterion is position error at most 3 mm, finite measurements and joint targets, legal arm targets, and completed preparation. The main

source-exchange panel uses position error, preserving the paired comparison across both damping conditions and all three requests.

For each of the 20 pairs, position errors are averaged over its six calls before taking swapped-minus-correct differences. The figure reports their mean and a paired 95% bootstrap interval. Correct and exchanged source conditions give aggregate position errors of 0.882 and $1.166\,\mathrm{mm}$, respectively. The experiment measures how changing retained source information affects the fixed response interface under the stated conditions. It is a separate recorded cohort from the fresh-process three-task evaluation.

E Execution Substitution and Calibration

E.1 Fixed response objects and execution conditions

The execution study uses the development-selected PAP-Affine model and 24 receiving states per task. For each receiver, the source packet is chosen from the first two metadata-ordered development packets for that task by nearest initial public context. The resulting u is retained throughout the operations. Three requested planar displacements, $(-3, 12)$, $(0, 18)$, and $(3, 21)\,\mathrm{mm}$, give 72 calls per condition. All other requested response components are zero. The selector uses the expanded 81-candidate grid and Eq. 12.

Conditions use paired restoration of the recorded receiving state. The measured outcomes are scored with the $5\,\mathrm{mm}$, 5°, terminal-speed, and legal-joint criteria in Appendix D. The main table includes all three tasks and each reported condition. This execution cohort has its own source selection and reset protocol; it is distinct from the independent-process Query–Invoke cohort.

E.2 Substituting temporal realization

Let $t \in \{1, \dots, 16\}$ index execution steps and $a_t = \min(t/12, 1)$. The desired planar TCP path is $p^{\mathrm{TCP}}_{0,xy} + R_\phi \xi^\star \rho(a_t)$, with fixed TCP height. We compare

$$\rho_{\mathrm{linear}}(a) = a, \qquad \rho_{\mathrm{smooth}}(a) = a^2(3 - 2a), \qquad \rho_{\mathrm{front}}(a) = \sqrt{a}. \tag{13}$$

All profiles reach the same requested displacement by step 12 and hold it for four steps. The PAP, goal, and selected spatial intervention are fixed within this comparison. Recorded desired TCP positions, measured TCP poses, joint targets, and object states make the changed realization and its physical outcome observable.

The temporal profile is an execution setting, not an additional input to the response head. Request satisfaction is evaluated from the measured object response after substitution. The retained prediction provides a reference for checking the measured response. This distinguishes retaining a response object from establishing its accuracy under a changed timing condition.

E.3 Calibrating a displacement-scale mismatch

A multiplicative gain $\gamma = 0.55$ is applied to commanded TCP travel. Calibration uses two reset-state strokes with requested forward displacements $d_1 = 20\,\mathrm{mm}$ and $d_2 = 30\,\mathrm{mm}$. Let m_1, m_2 be their measured forward TCP displacements in the local frame. The estimated gain and corrected command are

$$\widehat{\gamma} = \mathrm{clip}\left(\frac{m_2 - m_1}{d_2 - d_1}, 0.2, 1.5\right), \qquad \xi^{\mathrm{cmd}} = \frac{\xi^\star}{\widehat{\gamma}}. \tag{14}$$

The physical execution still applies γ; its effective requested travel is $\gamma\xi^\star/\widehat{\gamma}$. The gain estimator reads the two requested lengths and measured TCP displacements, not the injected value. Differencing the two strokes removes a common additive displacement offset under the linear gain model. Calibration costs 32 control steps, after which each evaluated call uses the same 16-step horizon as its reference condition. Calibration strokes start from reset states rather than altering the initial object state of the evaluated call.

The PAP, response model, request, and intervention-selection procedure remain fixed. Object motion is used to evaluate whether the corrected realization satisfies the request. For the reported conditions,

successful calls change from 0/72 under mismatch to 59/72 on PokeCube and 72/72 on RollBall; the corresponding reference counts are 60/72 and 72/72.

E.4 Two measurement interfaces and protocol interpretation

The tool residual $r_{\text{tool}} = \xi^{\text{obs}} - \xi^{\star}$ checks intervention realization. The response residual $r_{\text{response}} = y^{\text{obs}} - F_{\mathcal{P}}(h, \xi^{\star})$ checks the predicted physical consequence, using wrapped yaw differences where appropriate. Tool measurements determine the tested gain correction; object measurements evaluate the resulting effect. A response residual can reflect context, model prediction, or changed execution conditions, so these interfaces provide distinct checks rather than a unique causal diagnosis.

The recorded execution study uses simulator-state restoration between conditions. Subsequent repeat and branch-order checks on that collector found order sensitivity despite agreement in restored visible state. The execution results therefore describe the recorded paired-restoration protocol. The prospective three-task Query–Invoke evaluation in Appendix D uses independently prepared fresh processes. These results are reported as separate studies, with their respective reset procedures preserved.

F Real-Robot Setup and Measurement

F.1 Apparatus and tabletop arrangement

The hardware experiments use a Franka Emika Panda with a parallel gripper for PushT and PokeCube. A wooden board provides the planar workspace. A fixed RealSense camera observes the scene. Figure 7 shows the PushT arrangement, including the T-shaped object, robot contact region, and marked tabletop target.

The Panda base defines the robot world-frame origin. Tabletop markings provide physical references for placing the object and reading its motion. Before each trial, the object's geometric center is aligned with its marked starting location: the cube center for PokeCube and the geometric center of the T-shaped object for PushT. The requested forward direction is defined in the task plane. The source interaction and subsequent invocation use the same declared physical axes and units.

F.2 Physical measurement

Physical measurements record the object's planar displacement and orientation change after each movement, relative to the starting reference.

Let $\Delta p_i \in \mathbb{R}^2$ denote measured planar displacement and $\widehat{\Delta p}_i$ the corresponding PAP prediction. The reported position MAE is the mean planar Euclidean error,

$$\text{MAE}_{\text{pos}} = \frac{1}{N} \sum_{i=1}^{N} \left\| \widehat{\Delta p}_i - \Delta p_i \right\|_2 . \tag{15}$$

For Invoke, the measured forward displacement is the component along the marked request direction. Predictions, requested displacements, and physical measurements are compared in the same planar coordinates and converted to millimetres for reporting.

F.3 Arrangement of Query and Invoke trials

Real source interactions instantiate PAPs through the frozen feature pipeline and response model. Query evaluates predicted responses to subsequent interventions, with 12 trials per task. Invoke uses the retained response interface to select interventions for requested forward object displacements of 5 and $10\,\text{mm}$. Each task and request is evaluated over 20 trials, with the object returned to its starting reference between trials.

Invocation success. A real-robot invocation is successful when both physical criteria are satisfied at the end of the movement: the Euclidean distance between measured and requested planar displacement is at most $5\,\text{mm}$, and the absolute wrapped difference between measured and requested

yaw change is at most 5°. Both errors compare the measured effect with the requested effect in the same physical frame. The reported success rate is the fraction of trials satisfying both criteria.

The resulting Query position MAE is $3.673\,\mathrm{mm}$ for PushT and $1.289\,\mathrm{mm}$ for PokeCube. For the 5 and $10\,\mathrm{mm}$ requests, the recorded successful-call counts are 10/20 and 12/20 for PushT and 13/20 and 12/20 for PokeCube. Mean measured forward displacement is 3.229 and $7.978\,\mathrm{mm}$ for PushT and 3.872 and $8.313\,\mathrm{mm}$ for PokeCube, respectively. The main figure displays these counts as percentages and rounds the Query errors to two decimal places. The physical setup and physical measurement procedure are shared across both request magnitudes.