arXiv:2606.19728v1 [cs.RO] 18 Jun 2026

Bidirectional Tutoring for Developmental Motor Learning in Robots: Co-Developed Interaction Dynamics Support Stable Learning

Rui Fukushima, Jun Tani

***Abstract*—Infants are well known to develop their motor skills through dense interaction with caregivers. Although such social interaction is crucial for human development, motor-skill learning in robots is often treated as a unidirectional process in which robots passively receive demonstrations from tutors. This overlooks a key property of social interaction: it is inherently bidirectional, with tutor and learner dynamically adapting to each other. In such interactions, the robot's past experiences may function as prior constraints that shape the dynamics of their co-developed trajectories. We hypothesize that bidirectional tutoring allows such constraints to guide the formation of consistent behavioral patterns that preserve behavioral coherence and support generalization, whereas unidirectional interaction lacks such constraints and leads to broader, less consistent behavioral patterns. To examine this hypothesis, we conducted two experiments with a physical humanoid robot performing an object manipulation task: one involving human-robot interaction and another employing an AI tutor interacting with the real robot through an adaptive intervention mechanism designed to examine whether similar effects would emerge under more controlled conditions. We implement the developmental learning framework using a free-energy-principle-based neural network extended with generative replay, which supports stable sequence-by-sequence learning from single tutored episodes. Across both settings, bidirectional tutoring fostered consistent behaviors and stage-wise generalization, while the robot gradually required less tutor guidance. These results suggest that bidirectional tutoring, as an embodied and socially grounded approach, provides an effective scaffold for developmental motor learning in robots.**

***Index Terms*—Developmental robotics, Motor learning, Human-robot interaction, Humanoid robots, Free energy principle.**

I. Introduction

Infants acquire motor and other cognitive skills through continuous and mutual interaction with caregivers [1]–[3]. Such early social interactions are widely regarded as foundational to human developmental learning, where skills gradually emerge through guided embodied experiences [4]–[6]. This socially grounded nature of development has been extensively discussed in developmental psychology, most notably Vygotsky's theory [7], [8], which emphasizes that human development is scaffolded through social interaction. Inspired by this view, developmental robotics has also sought to incorporate social interaction into artificial learning systems [9]–[12]. However, it remains unclear how mutual interaction between tutor and learner shapes the interaction dynamics themselves, and how these dynamics influence developmental motor learning in robots.

R. Fukushima and J. Tani are with the Cognitive Neurorobotics Research Unit, Okinawa Institute of Science and Technology Graduate University, Okinawa, Japan.

Corresponding author: J. Tani. E-mail: jun.tani@oist.jp

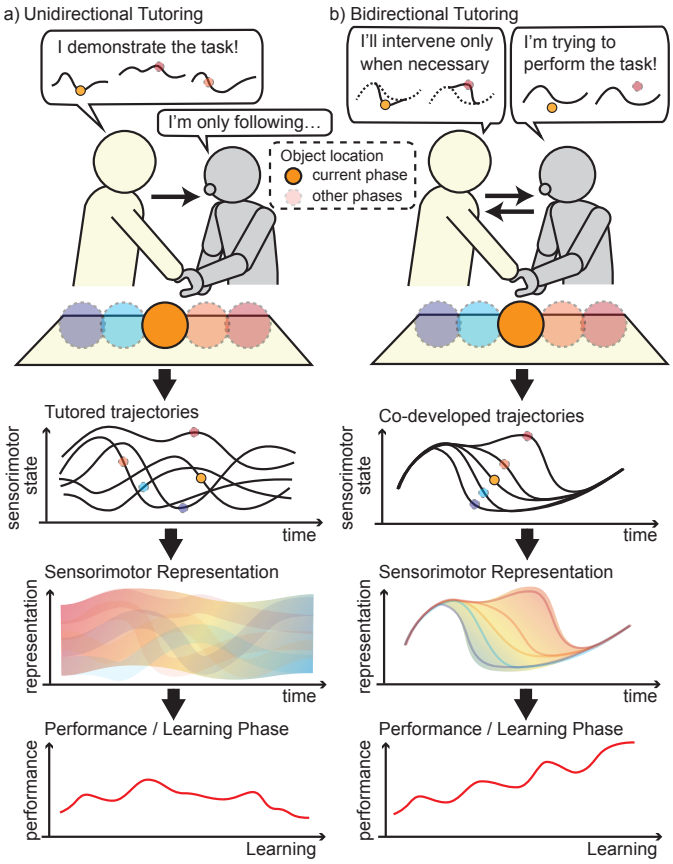

Fig. 1. **Conceptual diagram of unidirectional and bidirectional tutoring in developmental motor learning.** The upper interaction schematics illustrate how tutoring trajectories are generated in each condition, and the middle panels overlay the resulting trajectories across developmental phases. The speech bubbles contrast unidirectional tutoring, where the tutor demonstrates the task while the robot passively follows, with bidirectional tutoring, where the robot attempts the task and the tutor intervenes as needed. Colored circles in the workspace indicate object locations across developmental phases; orange denotes the current phase, and the other colors denote other phases. In the trajectory panels, colored dots indicate task-critical states required for successful task execution, such as reaching the object location. The same colors are used in the trajectory and representation panels to indicate the corresponding object locations. The lower panels illustrate how these trajectories may be organized in the model's sensorimotor representation space. **a** In unidirectional tutoring, tutor-provided demonstrations are illustrated as dispersed trajectories and as a poorly organized representation. **b** In bidirectional tutoring, co-developed trajectories are illustrated as more aligned trajectories and as a more coherently organized representation.

In robot motor learning, socially mediated skill acquisition is often formulated as learning from demonstration (LfD) or imitation learning (IL), where a robot acquires skills from tutor-provided examples [13]–[15]. A representative tutoring approach is kinesthetic teaching, in which a tutor physically

guides the robot through task-relevant movements [16]–[18]. Although kinesthetic teaching provides an intuitive way to demonstrate movements, it is often treated as a unidirectional process, in which the robot passively receives tutor-provided demonstrations [19]–[21]. In such settings, new tutoring experiences are introduced without considering how the robot's ongoing behavior and developing internal dynamics may influence the demonstrated trajectories. Such a view may overlook how tutor–robot interaction dynamics shape tutoring experiences, including the structure of the resulting trajectories, and how these experiences in turn affect learning and generalization. Previous studies have shown that learning depends not only on whether individual demonstrations succeed, but also on the consistency of sensorimotor patterns across demonstrations. In particular, coherent sensorimotor patterns support stable behavioral representations, whereas poorly structured or inconsistent demonstrations can hinder learning and generalization [22]–[25]. This concern may become particularly relevant in developmental learning, where new tutoring experiences may be shaped through interaction with the robot's prior learning rather than introduced independently of its developing sensorimotor dynamics.

Building on this view, we hypothesize that bidirectional tutor–robot interaction supports stable developmental motor learning by preserving behavioral coherence across developmental phases. Fig. 1 provides a conceptual overview of this hypothesis. In unidirectional tutoring (Fig. 1-a), tutor demonstrations are provided independently of the robot's behavior. Consequently, even when tutoring demonstrations satisfy task-critical states shown in Fig. 1, such as reaching the object and placing it at the target location, their detailed trajectories can still differ in shape and timing due to motor redundancy and natural variability across executions [26]–[28]. In this way, unidirectional tutoring may introduce successful but poorly structured demonstrations across developmental phases, which can hinder the formation of coherent sensorimotor representations and limit generalization. In contrast, in bidirectional tutoring (Fig. 1-b), tutor intervention is closely coupled with the robot's ongoing behavior. Because this behavior reflects the robot's previously acquired sensorimotor dynamics, new trajectories are shaped jointly by tutor guidance and the robot's prior learning. As a result, co-developed trajectories can remain compatible with previously learned patterns while still allowing tutor intervention in new situations.

To examine this hypothesis, we conducted developmental learning experiments with a physical humanoid robot under two tutoring modes: *unidirectional*, in which the robot passively receives guidance (Fig. 1-a), and *bidirectional*, in which the tutor and robot co-develop behavior through real-time interaction (Fig. 1-b). We first performed a proof-of-concept experiment with a human tutor, conducted by one of the experimenters, who physically interacted with the robot. We then introduced an AI tutor to assess whether the observed effects of bidirectional tutoring would also emerge under a more controlled tutoring policy setting. The framework is implemented using a neural model based on the free energy principle (FEP) [29]–[31], specifically the Predictive coding-inspired Variational Recurrent Neural Network (PV-RNN) [32]. The same PV-RNN framework is also used to implement the AI tutor. As a framework for developmental learning, we extended PV-RNN with a generative replay mechanism to enable stable sequence-by-sequence learning without storing previous tutored sequences.

The main contributions of this study are summarized as follows:

- We propose bidirectional tutoring as a developmental motor learning framework in which tutor intervention is coupled with the robot's ongoing behavior, allowing tutoring trajectories to be co-developed through tutor–robot interaction.
- We demonstrate, through human- and AI-tutoring experiments with a real robot, that bidirectional tutoring yields more coherent tutored trajectories and supports more stable phase-wise learning than unidirectional tutoring.
- We extend PV-RNN with generative replay to support stage-wise learning from single tutored sequences while mitigating catastrophic forgetting.

II. Background

A. From Demonstration to Interactive Motor Learning

Conventionally, kinesthetic teaching has been treated as a one-way process in which a robot passively receives tutor-provided demonstrations; for example, the robot's motors may be inactivated during tutoring [19]–[22], [33]. Such settings correspond to unidirectional interaction, where adaptation is not required from either the tutor or the robot. In contrast, several studies have explored more interactive forms of kinesthetic teaching, allowing the robot to generate motion under compliant control during tutoring [33]–[36]. More recently, Matsumoto et al. [37] advanced this direction by enabling online adaptation of robot motion in response to human physical guidance within an FEP-based active inference framework. In this framework, robot motion and tutor guidance were coupled during the acquisition of newly tutored episodes for incremental learning, marking a shift toward bidirectional tutoring. However, these studies did not directly examine how the bidirectional tutor-robot interaction shapes the interaction dynamics themselves, and how these dynamics, in turn, influence developmental motor learning. In contrast, the present study directly compares unidirectional and bidirectional tutoring to examine how the form of interaction shapes tutored trajectories and affects phase-wise developmental motor learning.

B. Free Energy Principle

The FEP provides a unifying framework for perception, action, and learning in biological systems [29]–[31]. Under this principle, agents minimize a statistical quantity called variational free energy, which can be expressed as an upper bound of the surprise or negative logarithm of marginal likelihood, $-\ln p(X)$, where X denotes the sensory observation.

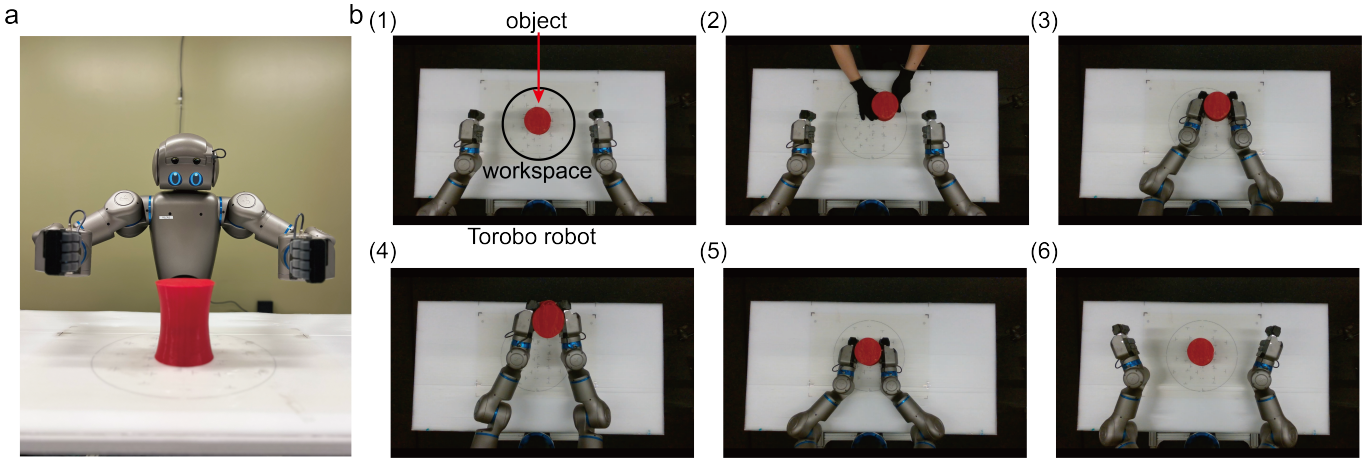

Fig. 2. **Overview of experimental setups and task settings a** Torobo humanoid robot. **b** Example of a successful task sequence: 1) A robot begins in a home posture with a red cylindrical object placed at the center of the workspace indicated by a black circle; 2) the experimenter moves the red object to a random position; 3) the robot begins to reach for and grasp the object; 4) the robot lifts up the object; 5) the robot places and releases the object at the original location 6) the robot returns toward its initial posture.

The variational free energy can be further decomposed into two components as follows:

$$\underbrace{-\ln p(X)}_{\text{surprise}} \leq \underbrace{\int q(z|X) \ln \frac{q(z|X)}{p(X,z)} dz}_{\text{variational free energy}} \quad (1)$$

$$= \underbrace{D_{\mathrm{KL}}[q(z|X)||p(z)]}_{\text{complexity}} - \underbrace{\mathbb{E}_{q(z|X)}[\ln p(X|z)]}_{\text{accuracy}} \quad (2)$$

The first term represents complexity, penalizing deviation of the approximated posterior distribution $q(z|X)$ of the latent variable z from the prior distribution $p(z)$. The second term represents accuracy, rewarding agreement between sensory prediction and observation.

The essential benefits for deployment of the FEP are twofold. First, its probabilistic formulation naturally handles the noise and stochasticity that are ubiquitous in embodied intelligent systems, as well as the subtle fluctuations arising from intentional conflicts during bidirectional interactions. Second, and more importantly, the FEP provides a mechanism for online inference by minimizing the mismatch between top-down priors and bottom-up posteriors. This allows the robot to continuously adjust its behavior in response to tutor intervention during real-time interaction [37], [38].

III. Methods

In this section, we present our approach to investigating developmental motor learning under unidirectional and bidirectional interaction. We first describe the experimental setup and experimental procedure, and then introduce the computational framework underlying the two tutoring modes.

A. Experimental Setup

Task Setting: To examine how different tutoring modes affect developmental motor learning, we designed a simple object manipulation task with a humanoid robot, Torobo (Tokyo Robotics) (Fig. 2-a). The objective of the task is to reach and grasp a cylindrical red object approximately 12 cm in diameter placed at an arbitrary position within a 36 cm in diameter workspace, and then return it to the original position (Fig. 2-b).

Torobo Humanoid Robot: The robot is equipped with a custom compliance module that allows humans to back-drive its joints during physical interaction. It has 6 joints for each arm and 2 joints for the torso. In the present experiment, the robot was controlled in Cartesian task space using a custom inverse kinematics module. The robot's proprioceptive sensations corresponded to the Cartesian poses of both end effectors, consisting of their positions and orientations. This proprioceptive representation contained 14 values in total: for each arm, 3 positional components (x, y, z) and 4 quaternion components (q_1, q_2, q_3, q_4) representing orientation. The visual sensation was represented by the two-dimensional center position of the object, extracted from overhead-camera images using an object-tracking program. Full pixel images were not used to simplify model learning and visual processing. All sensory and motor signals were sampled and processed at 10 Hz. Accordingly, each task episode consisted of a sequence of length T=650, corresponding to 65 seconds.

B. Experimental Procedure and Evaluation Metrics

In this study, we compared two tutoring modes: unidirectional tutoring (Fig. 3-a) and bidirectional tutoring (Fig. 3-b). In unidirectional tutoring, robot action generation was disabled, and the robot was set to gravity compensation mode so that the tutor could physically guide it through a successful task sequence. In bidirectional tutoring, the robot generated actions based on its own predictions under compliance mode, and the tutor intervened to correct the robot's behavior as necessary (Fig. 3-c).

Tutoring–Training Phase: The experiments progressed in a stage-wise manner through repeated tutoring–training phases. Each phase consisted of a single tutoring episode, followed by model training using the resulting tutored sequence. This

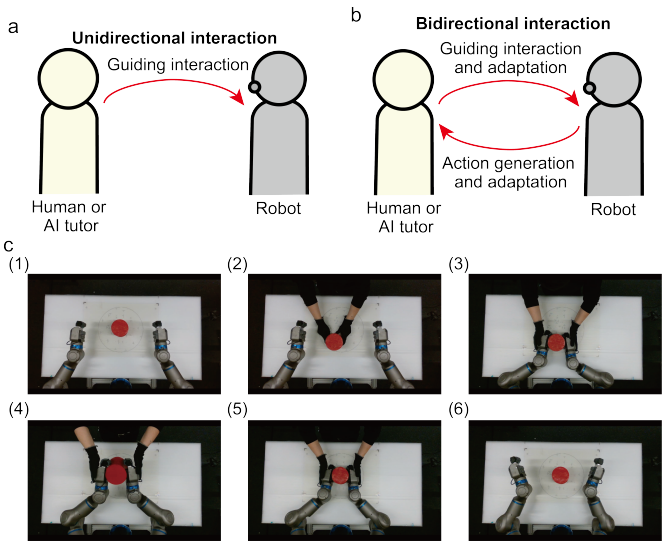

Fig. 3. **Overview of tutoring setups a** Diagram of unidirectional tutoring, where the human or AI tutor provides one-way guidance to the robot. **b** Diagram of bidirectional tutoring, where the robot generates actions based on its own predictions while the tutor intervenes as needed. **c** Example of a human-robot tutoring sequence. 1-2) The object is moved to a random position; 3-5) the tutor provides physical guidance for the robot to reach for, grasp, lift, and return the object to the center; 6) the robot returns toward its initial posture.

process was repeated across ten phases. Phase 1 served as a common baseline for both conditions. Because the robot started from scratch and had not yet acquired any task-related behavior, bidirectional interaction was not feasible at this stage. Therefore, tutoring in Phase 1 was performed only in the unidirectional mode, and the resulting trajectory was used to initialize both learning conditions.

To ensure a statistically fair evaluation, we considered three sources of variability: the object positions and their learning-order across phases, human tutoring, and network stochasticity. To account for the first two, we repeated the same experimental procedure across three independently sampled object-position sets (Sets A–C). Each set consisted of 10 positions corresponding to the 10 developmental phases. To account for network stochasticity, we adopted the following multiple-seed training and selection procedure: (1) In Phase 1, the model was trained on the initial tutored trajectory using 15 different random seeds, accounting for both parameter initialization and stochastic sampling. (2) The trained models were sorted by their final loss, and the median-loss model was selected to avoid bias toward extreme outcomes. (3) The selected model was further trained with the newly obtained tutored sequence in 15 separate runs, each using a different random seed. (4) The trained models were again sorted by the final loss, and the median-loss model was selected for the next phase. Steps 3 and 4 were repeated through Phase 10. This procedure was applied independently to the experiment conducted with each object-position set (Sets A–C).

Testing Phase and Evaluation Metrics*:* In the evaluation phase, the robot was expected to autonomously generate its own behavior without tutor assistance (Fig. 2-b). To evaluate generalization performance, a separate test set consisting of 10 randomly sampled object positions was prepared, and each model was tested on this set at every stage ($N_{\text{test}} = 10$).

We evaluated model performance using the success rate, defined as $\frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \mathbb{I}^i$, where $\mathbb{I}^i$ denotes the success indicator for the i-th test episode. The success indicator is defined as follows:

$$\mathbb{I}^i = \begin{cases} 1 & \text{if the episode was successful,} \\ & d^i_{\text{reach}} < T_{\text{err}} \text{ and } d^i_{\text{place}} < T_{\text{err}}, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Here, an episode was regarded as completed successfully if the robot finished the full manipulation sequence and released the object within the given episode length. Episodes in which the robot dropped the object or failed to complete the sequence within the episode length were counted as failures. The spatial error threshold T_{err} was introduced to evaluate the precision of the resulting behavior, so that success required not only task completion but also sufficient precision in reaching the object and placing it at the designated return location. This was necessary because the robot occasionally pushed the object with one hand before grasping it with both hands, or returned it to a position displaced from the designated return location. Therefore, we computed two spatial errors for each episode: the reaching error d^i_{reach} and the placement error d^i_{place}. An episode was counted as successful only when both errors were smaller than the predefined threshold T_{err}.

C. Computational Framework

In this section, we describe the computational framework used in this study, namely the Predictive coding–inspired Variational Recurrent Neural Network (PV-RNN). PV-RNN is grounded in the FEP, allowing the robot to continuously infer posterior latent variables by minimizing variational free energy, thereby supporting real-time behavioral adjustment. In the following section, we first introduce the model overview and training configuration, and then describe the online inference procedure used for real-time posterior inference and action generation, and finally present the generative replay method used in this study to support stage-wise developmental learning.

Model Overview and Training Configuration*:* As illustrated in Fig. 4-a, the network consists of two main components: the PV-RNN layer and the output layer. At each time step t, the prediction $\bar{\boldsymbol{x}}_t$ is generated from the deterministic hidden state $\boldsymbol{d}_t$ and the stochastic latent variable $\boldsymbol{z}_t$. $\boldsymbol{a}_t$ is an adaptive vector used to infer the approximate posterior in each time step [32]. Here, these adaptive vectors are assigned in a sequence-specific manner, i.e., each training sequence is associated with its own set of adaptive vectors. The output layer is implemented as a fully connected layer, which transforms the deterministic latent state $\boldsymbol{d}_t$ into sensorimotor prediction $\bar{\boldsymbol{x}}_t$. (See Appendix for the detailed computations.)

The objective function of the model is to minimize the variational free energy, equivalently maximizing the evidence

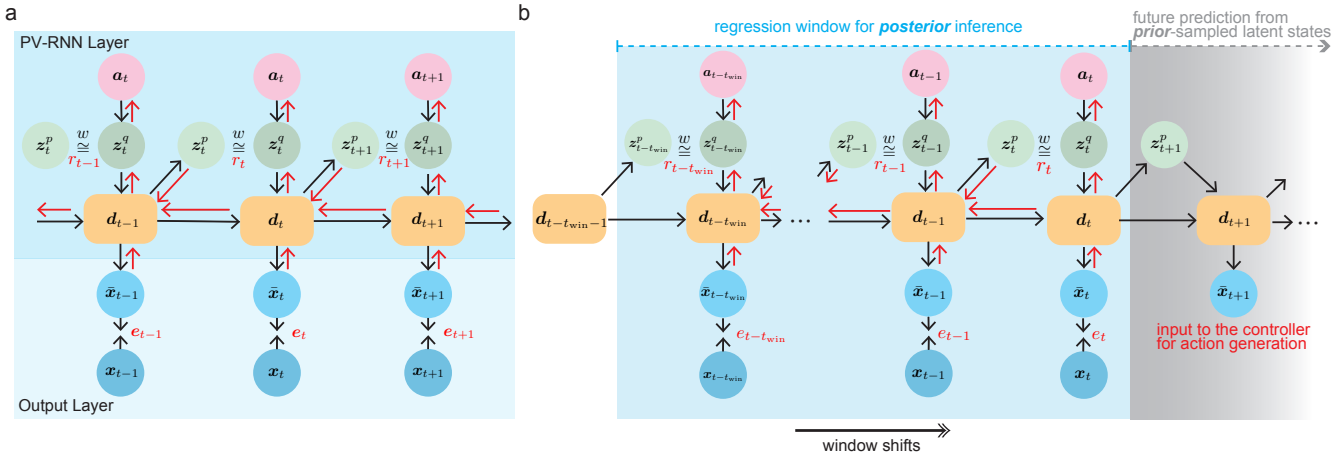

Fig. 4. **Illustration of the computational flow of the PV-RNN during training and online inference. a** Training process. The model generates predictions over time through recurrent computation. At each time step, the reconstruction error e_t is computed from the prediction $\bar{\boldsymbol{x}}_t$ and observation $\boldsymbol{x}_t$, while r_t denotes the Kullback–Leibler divergence between the posterior $\boldsymbol{z}_t^q$ and prior $\boldsymbol{z}_t^p$ latent distributions. **b** Online inference with a shifting regression window. Posterior estimates are updated within the window through error regression, and the resulting one-step-ahead prediction $\bar{\boldsymbol{x}}_{t+1}$ is used for action generation. As time advances, the regression window shifts forward, and the same procedure is repeated. Black arrows indicate forward computation, whereas red arrows indicate error backpropagation.

lower bound (ELBO) $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$ defined as follows:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \sum_{t=1}^{T} \Bigg\{ \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_t)} \left[\ln p_{\boldsymbol{\theta}}(\boldsymbol{x}_t \mid \boldsymbol{d}_t) \right]}_{\text{Accuracy: } e_t} - w \underbrace{D_{\mathrm{KL}} \left[q_{\boldsymbol{\phi}}(\boldsymbol{z}_t) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{z}_t \mid \boldsymbol{d}_{t-1}) \right]}_{\text{Complexity: } r_t} \Bigg\} \tag{4}$$

Here, $\boldsymbol{\theta}$ denotes the parameters of the generative model, which predicts sensory observations $\boldsymbol{x}_t$, $\boldsymbol{\phi}$ denotes the parameters of the inference model, which approximates the posterior distribution of latent variables given sensory observations. In the present PV-RNN model, $\boldsymbol{\theta}$ includes network weights, biases, and trainable initial hidden states $\boldsymbol{h}_0$, while $\boldsymbol{\phi}$ is realized as a set of adaptive vectors. w is a hyperparameter called meta-prior, which balances the accuracy term and complexity term in the variational free energy [32].

The network was trained through backpropagation through time (BPTT) [39], and the optimization proceeds by gradient descent using the Adam optimizer with parameters α=0.01, β_1=0.9, β_2=0.999 [40]. The network parameters were set as follows: N_d=60, N_z=1, τ=8, and w=0.01. Here, N_d and N_z denote the number of deterministic $\boldsymbol{d}$ and probabilistic $\boldsymbol{z}$ unit, respectively. The parameter τ defines the time constant for hidden-state computation, and w corresponds to the meta-prior. The number of training epochs N_{epoch} in each phase was set to 30,000.

Online Inference and Action Generation: During both the tutoring and testing phases, PV-RNN performed online inference through an error regression (ER) process with a shifting window [32], [41]. The overall computational flow is illustrated in Fig. 4-b (see Appendix for detailed computations). In this process, the network dynamically adjusted its inference parameters $\boldsymbol{\phi}$ based on the mismatch between sensory observations and predictions, while keeping the generative parameters $\boldsymbol{\theta}$ fixed. Specifically, $\boldsymbol{\phi}$ was repeatedly optimized by minimizing the following variational free energy:

$$\mathcal{L}(\boldsymbol{\phi}) = \sum_{t'=t-t_{\mathrm{win}}}^{t} \Bigg\{ \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{t-t_{\mathrm{win}}:t'})} \left[\ln p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t'} \mid \boldsymbol{d}_{t'}) \right]}_{\text{Accuracy: } e_{t'}} - w \underbrace{D_{\mathrm{KL}} \left[q_{\boldsymbol{\phi}}(\boldsymbol{z}_{t'}) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{z}_{t'} \mid \boldsymbol{d}_{t'-1}) \right]}_{\text{Complexity: } r_{t'}} \Bigg\} \tag{5}$$

ER was performed using BPTT within a regression window of length t_{win} for N_{itr} optimization steps. In this study, we set t_{win}=100 and N_{itr}=50. The same Adam optimizer settings as in training were used. After optimization, the one-step-ahead prediction $\bar{\boldsymbol{x}}_{t+1}$ was generated, and its proprioceptive components were sent to the robot controller for action generation. The regression window was then shifted forward as time advanced, and the same procedure was repeated.

PV-RNN-Based Generative Replay for Stage-Wise Learning: To support stage-wise learning, we developed a generative replay mechanism tailored to the PV-RNN learning framework [42]–[44]. In this process, replay sequences were generated by sampling latent states from the prior distribution of the trained generative model while recursively updating the recurrent dynamics forward in time. Because these sequences reflect previously acquired sensorimotor patterns, replaying them during later training helps mitigate catastrophic forgetting without storing the previous training data [45], [46].

In the present stage-wise learning setting, the model needed to incorporate a newly tutored sequence at each phase while preserving sensorimotor patterns acquired in previous phases. Because only one new tutored sequence was available at each phase, standard mini-batch training with replay samples could easily result in either overfitting or forgetting. Too many replay samples could bias training toward previously acquired sensorimotor patterns, whereas too few replay samples could

Algorithm 1 Generative replay with PV-RNN

Require: New tutored sequence $\boldsymbol{X}_{\text{new}} = \{\boldsymbol{x}_t\}_{t=1}^{T}$ and its inferred adaptive vectors $\boldsymbol{A}_{\text{new}} = \{\boldsymbol{a}_t\}_{t=1}^{T}$

1: Generate replay sequences from prior-sampled latent states of the current model, and obtain the corresponding prior parameters:

$$\bar{\boldsymbol{X}}_j = \{\bar{\boldsymbol{x}}_{t,j}\}_{t=1}^{T},\ \{(\boldsymbol{\mu}^p_{t,j}, \boldsymbol{\sigma}^p_{t,j})\}_{t=1}^{T} \qquad j = 1, \ldots, N_{\text{replay}}$$

2: Initialize the corresponding adaptive vectors from the prior parameters:

$$\boldsymbol{a}^{\mu}_{t,j} \leftarrow \operatorname{arctanh}(\boldsymbol{\mu}^p_{t,j}), \quad \boldsymbol{a}^{\sigma}_{t,j} \leftarrow \operatorname{logit}(\boldsymbol{\sigma}^p_{t,j})$$

$$\boldsymbol{A}_j = \{(\boldsymbol{a}^{\mu}_{t,j}, \boldsymbol{a}^{\sigma}_{t,j})\}_{t=1}^{T} \qquad j = 1, \ldots, N_{\text{replay}}$$

3: $\mathcal{M} \leftarrow \{(\bar{\boldsymbol{X}}_j, \boldsymbol{A}_j)\}_{j=1}^{N_{\text{replay}}}$

4: **for** N_{epoch} **do**

5: $\quad \mathcal{B} \leftarrow \{(\boldsymbol{X}_{\text{new}}, \boldsymbol{A}_{\text{new}})\} \cup \{(\bar{\boldsymbol{X}}_k, \boldsymbol{A}_k)\}_{k=1}^{N_{\mathcal{B}}-1} \sim \mathcal{M}$

6: $\quad$ Update $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ using one epoch of gradient descent on $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathcal{B})$

7: **end for**

provide insufficient information about those patterns and lead to forgetting. To address this issue, we generated a replay buffer containing many replay-sequences, but used randomly sampled small subset of them at each training epoch. This allowed the model to repeatedly revisit past sensorimotor patterns without overwhelming the newly tutored sequence. Algorithm 1 provides the pseudo-code for this generative replay procedure used in this study. After each tutoring episode, the resulting tutored sequence was used as the new training sequence $\boldsymbol{X}_{\text{new}}{=}\{\boldsymbol{x}_t\}_{t=1}^{T}$. Before training on this sequence, a replay buffer $\mathcal{M}$ was constructed from N_{replay} replay sequences generated from the current trained model. In addition, because PV-RNN requires sequence-specific adaptive vectors, adaptive vectors were prepared for both the new tutored sequence and replay sequences. Although adaptive vectors are randomly initialized in standard PV-RNN training [32], random initialization was less stable in this incremental learning setting because it ignored sequence-specific latent information. We therefore initialized the adaptive vectors as follows. For the new tutored sequence, the adaptive vectors inferred during the tutoring session were used as $\boldsymbol{A}_{\text{new}}$. For each replay sequence, the adaptive vectors $\boldsymbol{A}_j$ were initialized using the sequence-specific stochastic parameters $\{(\boldsymbol{\mu}^p_{t,j}, \boldsymbol{\sigma}^p_{t,j})\}_{t=1}^{T}$ obtained during its replay generation. This allowed both the new and replayed sequences to be initialized with adaptive vectors consistent with their corresponding latent dynamics before training. Here, $\operatorname{arctanh}(\cdot)$ and $\operatorname{logit}(\cdot)$ were used as inverse transformations of those used to compute the posterior parameters $\boldsymbol{\mu}^q$ and $\boldsymbol{\sigma}^q$, respectively (see Appendix). Finally, at each training epoch, the model is updated using the training batch $\mathcal{B}$ of size $N_{\mathcal{B}}$, which consisted of the new tutored sequence and a randomly sampled subset of replay sequences. In the present experiments, $N_{\text{replay}}{=}1023$ and $N_{\mathcal{B}}{=}8$ were used.

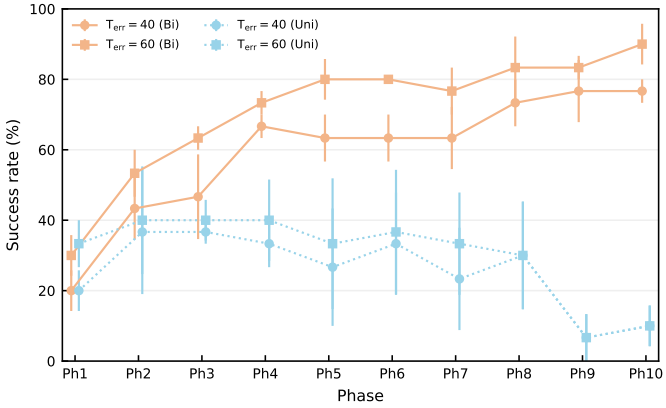

Fig. 5. **Success rate across developmental phases for the unidirectional (Uni) and bidirectional (Bi) tutoring conditions.** Data points represent the mean success rate across training sets A–C, and error bars indicate the standard error of the mean (SEM). Two spatial error thresholds are used, $T_{\text{err}}{=}40$ and 60.

IV. Human Tutor Experiment: Proof of Concept

As an initial proof of concept, we first conducted a developmental motor learning experiment in which a human tutor physically interacted with a humanoid robot. This experiment examined whether the proposed bidirectional tutoring process can arise in real physical human–robot interaction, where both tutor and robot are embodied agents coupled through physical contact and real-time sensorimotor feedback. Testing multiple human tutors in this setting is practically difficult because each participant would need to complete the full multi-stage procedure, requiring substantial experimental time. In the present proof-of-concept experiment, one of the experimenters served as the human tutor. For this reason, the human-tutoring experiment is positioned as an embodied demonstration of the proposed mechanism, which was further evaluated under more controlled conditions using an AI tutor in Section V.

A. Task Performance

First, we evaluated task performance across ten training phases by computing the success rate under two spatial error thresholds ($T_{\text{err}}{=}40$ and 60 pixels; 1 pixel$\approx$1.088 mm), where the smaller threshold indicates the stricter success criterion. Fig. 5 shows the phase-wise task performance for the unidirectional and bidirectional tutoring conditions. Overall, the bidirectional condition consistently achieved higher success rates than the unidirectional condition across phases under both error thresholds. In the bidirectional condition, the success rates increased progressively over developmental phases, indicating gradual improvement in task performance through stage-wise learning. In contrast, the unidirectional condition remained substantially lower overall and did not exhibit a clear improvement trend. Instead, performance plateaued in the early phases and declined in the later phases. The gap between the two conditions became particularly pronounced in the later phases, where the mean success rate in the bidirectional condition reached 90.0% under $T_{\text{err}}{=}60$ and 76.7% under $T_{\text{err}}{=}40$, whereas that in the unidirectional condition remained at 10.0% under both thresholds. These results indicate that bidirectional

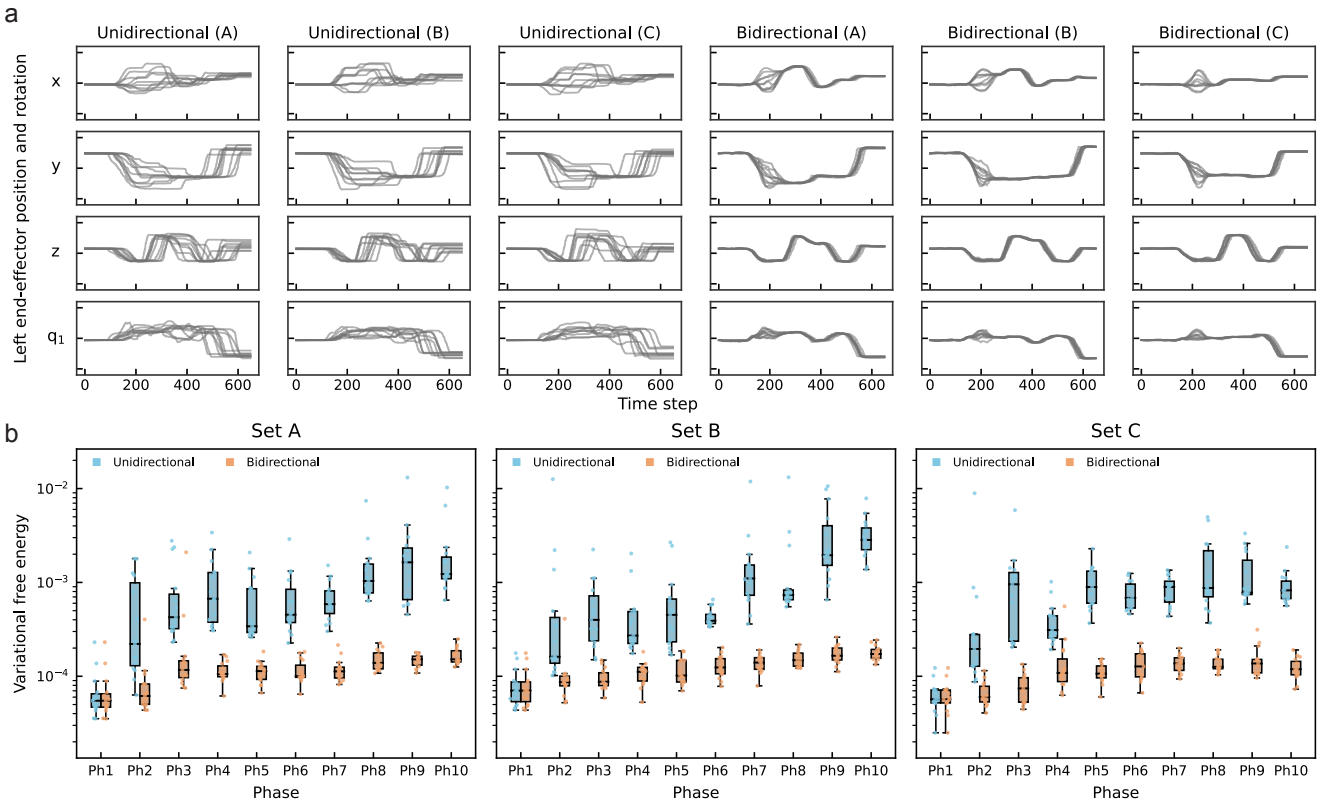

Fig. 6. **Qualitative and quantitative comparison of tutoring dynamics under the unidirectional and bidirectional conditions. a** Left end-effector position and rotation trajectories during tutoring, overlaid across all tutoring phases for each training set (A–C). Each trace corresponds to the tutored sequence at one phase. For rotation, only the first quaternion component (q_1) is shown, with the other components (q_2, q_3, and q_4) omitted for clarity. **b** Training loss, quantified as variational free energy, across developmental phases for each training set (A–C). Box plots show the distribution across random seeds. Statistical analysis was conducted on log-transformed values from Phases 2–10. Phase 1 was excluded from the statistical analysis because it served as a common baseline before the tutoring conditions diverged.

tutoring better supported stage-wise motor learning, whereas unidirectional tutoring showed limited improvement and a decline in later phases. Representative autonomous testing episodes are shown in Supplementary Video 1, illustrating the difference in task execution between the unidirectional and bidirectional conditions.

B. Qualitative and Quantitative Comparison

To better understand the factors underlying the difference in task performance, we examined the qualitative and quantitative characteristics of the tutored trajectories. In the present stage-wise learning procedure, each developmental phase introduced a new tutored trajectory that was used for subsequent model training. Therefore, the structure of these trajectories was expected to affect how compatible new tutoring experiences were with the model's previously acquired sensorimotor patterns. We first examined the spatiotemporal organization of the tutored trajectories across developmental phases. We then analyzed the training loss to assess whether differences in trajectory organization were reflected in the model's learning dynamics. To provide an intuitive view of the tutoring process, representative tutoring episodes are shown in Supplementary Video 2. The video highlights differences in how tutored trajectories were formed under the two tutoring modes and shows the robot's predictions, observations, internal states, and tutor-applied excess torque.

Spatiotemporal Structure of Tutored Trajectories: We first visualized the trajectories obtained during tutoring sessions across the ten developmental phases. Fig. 6-a overlays the left end-effector position and rotation trajectories obtained during tutoring across all developmental phases under unidirectional and bidirectional conditions. In the unidirectional condition, the trajectories exhibited substantial spatial and temporal variability across all dimensions. In contrast, in the bidirectional condition, the trajectories showed stronger spatiotemporal consistency across developmental phases. Importantly, this consistency was expressed differently depending on the position and rotation components. The x- and y-position trajectories showed spatial dispersion around t=200, corresponding to the reaching movement toward different object locations. This dispersion likely reflects task-critical adjustments, where the robot had to approach newly introduced object locations under tutor guidance. Outside these task-critical adjustments, the x- and y-position trajectories became more aligned in both shape and timing across phases. The vertical position trajectory (z) and rotation trajectory (shown here by the representative quaternion component q_1) also showed clear temporal consistency across phases. This consistency likely reflects the fact that these movement components required less adjustment and

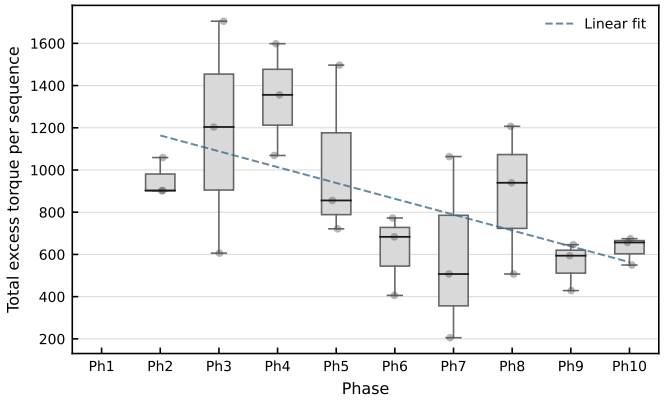

Fig. 7. **Changes in human tutor intervention during bidirectional tutoring.** Box plots show the total excess torque applied by the human tutor during each tutoring episode in the bidirectional condition. Data from learning sets A–C were pooled within each developmental phase. The dashed line indicates the linear trend estimated by linear regression across developmental phases. Total excess torque decreased over phases (Pearson's correlation, $r{=}-0.726$, $p{=}0.0268$; linear regression, $R^2{=}0.527$). No value is shown for Ph1 because this baseline phase did not include robot-to-tutor physical feedback, which was required to compute this measure.

could therefore rely more on sensorimotor patterns acquired in previous phases. Together, these observations suggest that bidirectional tutoring did not simply reduce variability uniformly. Rather, the resulting trajectories reflected both tutor-guided adjustments to newly introduced object locations and movement patterns previously acquired by the robot.

Training Loss and Trajectory Compatibility: We next examined the training loss, quantified as the variational free energy, across developmental phases. In the present context, the loss value can be interpreted as an index of how compatible the tutored trajectories were with the model's previously acquired sensorimotor dynamics. Higher loss suggests that the trajectories were harder to predict and integrate with prior sensorimotor patterns, likely because they introduced greater variability or spatiotemporal inconsistency. Lower loss, in contrast, may indicate that the trajectories were easier to predict and integrate because they were more compatible with previously acquired sensorimotor dynamics.

As shown in Fig. 6-b, the bidirectional condition generally yielded lower training loss than the unidirectional condition across developmental phases and training sets. This difference remained significant in a linear model that accounted for developmental phase and training set (condition effect for bidirectional tutoring relative to unidirectional tutoring: $\beta{=}-1.818$, SE$=0.044$, $t{=}-41.01$, $p{<}0.001$). Although loss increased overall across phases in both tutoring conditions, this trend mainly reflects the cumulative learning procedure, in which a newly tutored sequence was added at each phase and expanded the range and complexity of sensorimotor patterns that the model had to represent. Overall, the lower loss in the bidirectional condition suggests that its tutored trajectories were more compatible with the model's previously acquired sensorimotor dynamics than those obtained under unidirectional tutoring.

Reduction of Tutor Intervention across Developmental Phases: In addition to the qualitative and quantitative characteristics of the tutored trajectories, we examined the total amount of tutor intervention across developmental phases in the bidirectional condition as an additional behavioral index of learning dynamics. As shown in Fig. 7, the total excess torque showed an overall decreasing trend across developmental phases. This reduction suggests that the robot gradually required less external guidance from the tutor across developmental phases. Together with the increased task success rate observed in Fig. 5, this decrease in tutor intervention suggests that the robot acquired task-relevant motor skills while becoming less dependent on tutor guidance.

V. AI TUTOR EXPERIMENT: CONTROLLED VALIDATION

The human-tutoring experiment provided an initial proof of concept that bidirectional tutoring can support stable developmental learning through the formation of consistent behavioral patterns. To examine this mechanism under more controlled conditions, we implemented an AI tutor that captured selected functional aspects of human tutoring. Using the same AI tutor in both conditions allowed us to examine the effect of interaction mode while keeping the tutoring policy fixed.

A. Training the PV-RNN-Based AI Tutor

We first describe how the AI tutor model was trained. The AI tutor was designed to provide a controlled tutoring policy for examining the effect of interaction mode while minimizing the direct influence of a particular human tutor. Therefore, instead of training the AI tutor on recorded human demonstrations, we generated the training trajectories computationally. Furthermore, these trajectories were designed to preserve the task-relevant structure while incorporating spatiotemporal variability. This avoided purely deterministic training examples and allowed the AI tutor to learn variability inherent in embodied motor behavior. To cover different object-location conditions and possible movement variations,

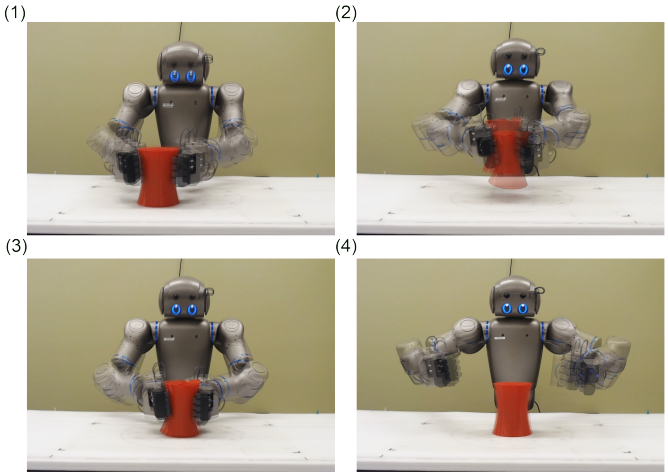

Fig. 8. **Example of variability in AI-tutor-generated trajectories under the same object-location condition.** Panels (1)–(4) show overlaid snapshots from three repeated executions at representative time points: (1) reaching and grasping at $t{=}200$, (2) lift-up at $t{=}400$, (3) put-down at $t{=}500$, and (4) returns toward its initial posture at $t{=}650$. The snapshots illustrate that the trained AI tutor generated task-relevant movements with variability, rather than producing a single deterministic trajectory.

we generated 555 trajectories and used them to train the AI tutor model. Details of the training-trajectory generation procedure are provided in Supplementary Material.

The AI tutor model was implemented using a PV-RNN model for two reasons. First, as a probabilistic recurrent model, PV-RNN is well suited for learning and generating stochastic time-series data [17], allowing the AI tutor to capture variability in the training trajectories. Second, the PV-RNN framework supports online inference, allowing the AI tutor to update its prediction in response to the robot's ongoing movement during bidirectional interaction. This online update was used to adjust tutor intervention, as described in the next section (Section V-B). The network parameters of the AI tutor model were set to N_d=100, N_z=15, τ=16, w=0.01, and N_{epoch}=100,000. Fig. 8 shows three repeated generations by the trained AI tutor under the same object-location condition, illustrating variable movement execution while preserving the task-relevant sequence structure. This trained AI model was used in the following AI-tutoring experiments.

B. Bidirectional Interaction between AI tutor and Robot

We next describe how the AI tutor interacted with the real robot and adjusted its intervention based on the robot's ongoing behavior. The AI tutor was not intended to reproduce full human tutoring behavior. Instead, it was designed as a controlled intervention policy that allowed us to isolate whether coupling the tutor intervention to the robot's ongoing behavior is sufficient to produce coherent developmental trajectories.

In the unidirectional condition, the AI tutor generated a tutoring trajectory based on its learned dynamics, and the robot passively followed this trajectory without autonomous action generation. In the bidirectional condition, by contrast, the robot generated its own intended action from its learned dynamics, while the AI tutor intervened as needed. To implement such bidirectional interactions, the AI tutor needed to determine when and how strongly to intervene based on the robot's ongoing behavior. We formulated this intervention process using the tutor's expected movement variability, so that tutor intervention could be adjusted according to the context of the ongoing movement. The following subsections describe this mechanism in four steps: 1) estimation of the tutor's expected movement variability, 2) variability-weighted deviation computation, 3) intervention-weight computation, and 4) joint-action generation.

1) Estimation of the Tutor's Expected Movement Variability: First, we estimated the tutor's expected movement variability for the upcoming movement. This quantity serves as a context-dependent tolerance for interpreting the deviation between the robot's intended action and the AI tutor's predicted action. Because the predicted variance of the prior latent distribution $(\boldsymbol{\sigma}^{p,\text{tutor}}_{t+1})^2$ represents uncertainty in the tutor's next-step prediction, we used it as a proxy for this expected variability. Specifically, we computed

$$\nu_{t+1} = \frac{1}{N_z}\sum_{k=1}^{N_z}(\boldsymbol{\sigma}^{p,\text{tutor}}_{t+1,k})^2 \tag{6}$$

where ν_{t+1} denotes the tutor's expected movement variability for the upcoming movement, computed as the mean predicted latent variance across the N_z stochastic units.

2) Variability-weighted Deviation Computation: Second, we computed the deviation between the robot's intended action and the AI tutor's predicted action relative to the expected variability estimated in Step 1. This allowed the same absolute action difference to be interpreted in a context-dependent manner: a deviation was treated as more significant when the tutor expected a narrow range of possible movements, and less significant when the tutor expected a broader range of possible movements. For each proprioceptive dimension, the variability-weighted deviation was computed as

$$\boldsymbol{\delta}_{t+1,i} = \frac{1}{\nu_{t+1}}\left|\bar{\boldsymbol{x}}^{\text{robot}}_{t+1,i} - \bar{\boldsymbol{x}}^{\text{tutor}}_{t+1,i}\right| \tag{7}$$

where $\bar{x}^{\text{robot}}_{t+1,i}$ and $\bar{x}^{\text{tutor}}_{t+1,i}$ denote the i-th proprioceptive dimensions of the robot's intended action and the AI tutor's predicted action, respectively.

3) Intervention Weight Computation: Third, we converted the variability-weighted deviation into an intervention weight. This mapping was designed to satisfy three practical requirements: the AI tutor should not intervene for deviations within a noise threshold, the intervention strength should increase gradually as the deviation becomes larger, and the resulting weight should remain bounded between 0 and 1. We implemented this mapping using a simple bounded function that satisfied these requirements as follows:

$$\tilde{\boldsymbol{w}}_{t+1,i} = \begin{cases} 1-\exp(-\lambda\boldsymbol{\delta}_{t+1,i}) & \text{if } \boldsymbol{\delta}_{t+1,i} > \epsilon_{\text{noise}}, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

where $\tilde{w}_{t+1,i}$ denotes the intervention weight for the i-th proprioceptive dimension. The parameter λ controls how rapidly the intervention weight increases, and ϵ_{noise} defines a noise threshold. In this study, we set λ=10 and ϵ_{noise}=1.0.

4) Joint-action Generation: Finally, the joint action was generated by combining the robot's intended action and the AI tutor's predicted action according to the intervention weight. For each proprioceptive dimension, the joint action was defined as

$$\bar{\boldsymbol{x}}^{\text{joint}}_{t+1,i} = (1-\tilde{\boldsymbol{w}}_{t+1,i})\bar{\boldsymbol{x}}^{\text{robot}}_{t+1,i} + \tilde{\boldsymbol{w}}_{t+1,i}\bar{\boldsymbol{x}}^{\text{tutor}}_{t+1,i} \tag{9}$$

where $\bar{\boldsymbol{x}}^{\text{joint}}_{t+1,i}$ denotes the i-th proprioceptive component of the generated joint action. Smaller values of $\tilde{w}_{t+1,i}$ keep the joint action closer to the robot's intended action, whereas larger values correspond to stronger tutor intervention, pulling the joint action toward the AI tutor's prediction.

Fig. 9 provides an intuitive illustration of how the proposed intervention mechanism operates through online inference. The figure contrasts two representative cases: a low-variability case, in which the tutor expects a narrow range of possible movements, and a high-variability case, in which the tutor expects a broader range of possible movements. These examples illustrate that the same robot–tutor action deviation can result in different interaction dynamics depending on the tutor's expected movement variability.

We next examined whether the advantage of bidirectional tutoring observed in the human-tutoring experiment also

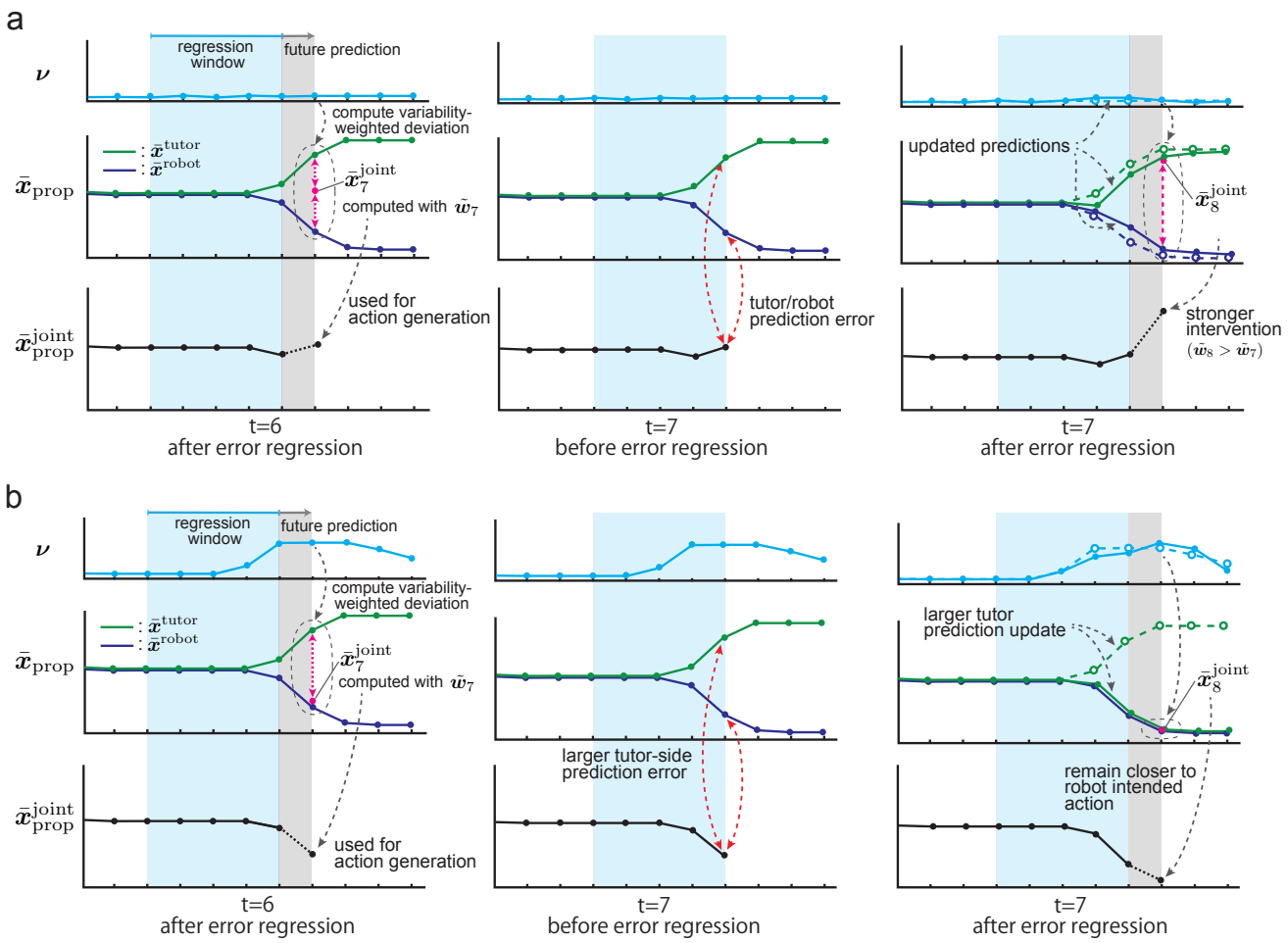

Fig. 9. **Schematic illustration of AI-tutor intervention through online inference in the bidirectional condition.** (a) Low expected movement variability case, where ν is small in the AI tutor model. (b) High expected movement variability case, where ν is large in the AI tutor model. At t=6 (left), after observing $\boldsymbol{x}_{1:6}$, the tutor and the robot generate their predicted actions, and the joint action is computed from these predictions. For a similar deviation between the robot's intended action and the tutor's prediction, the low-variability case yields a larger intervention weight $\tilde{\boldsymbol{w}}$, so the joint action is drawn closer to the tutor's intended action. In contrast, when ν is larger, the intervention weight becomes smaller, and the joint action remains closer to the robot's intended action. At t=7 before error regression (middle), prediction error is computed from the mismatch between the predicted trajectory $\bar{\boldsymbol{x}}_{3:7}$ and the observation $\boldsymbol{x}_{3:7}$. Because weaker intervention in the high-variability case makes the generated observation farther from the tutor's prediction, the tutor-side prediction error becomes larger than in the low-variability case. At t=7 after error regression (right), both predictions are updated based on their respective prediction errors. In the low-variability case, the next joint action $\bar{\boldsymbol{x}}_8^{\text{joint}}$ is pulled further toward the tutor's prediction because the updated intervention weight becomes larger ($\tilde{\boldsymbol{w}}_8 > \tilde{\boldsymbol{w}}_7$). In the high-variability case, the tutor prediction is updated more strongly, resulting in the joint action remaining closer to the robot's intended action.

emerges under this controlled AI-tutoring framework. To this end, we conducted the AI-tutoring experiment using the same overall experimental design as in the human-tutoring experiment and analyzed both task performance and tutoring dynamics.

C. Task Performance

We first evaluated task performance in the AI-tutoring experiment using the same success rate measure as in the human-tutoring experiment. Fig. 10 shows the success rate across developmental phases for the unidirectional and bidirectional conditions. Overall, the success rate increased during the early developmental phases in both conditions. However, subsequent performance trends differed between the two tutoring conditions. In the bidirectional condition, the success rate continued to increase and remained high in the later phases under both error thresholds. By contrast, in the unidirectional condition, the success rate improved up to the middle phases but declined thereafter. By the final phase, the mean success rate in the bidirectional condition reached 83.3% under T_{err}=60 and 66.7% under T_{err}=40. In contrast, the corresponding success rates in the unidirectional condition were 26.7% and 13.3%, respectively. These results indicate that the advantage of bidirectional tutoring observed in the human-tutoring experiment was also evident under the controlled AI-tutoring framework. Specifically, bidirectional tutoring shows higher and more stable task performance, whereas unidirectional tutoring shows limited improvement followed by a decline in later phases. A representative autonomous testing episode is shown in Fig. 11. In this example, the unidirectional condition failed to complete the task, whereas the bidirectional condition succeeded. This failure may reflect a mode-averaging-like effect, in which inconsistencies among accumulated tutoring experiences may lead to a poorly organized sensorimotor representation that no

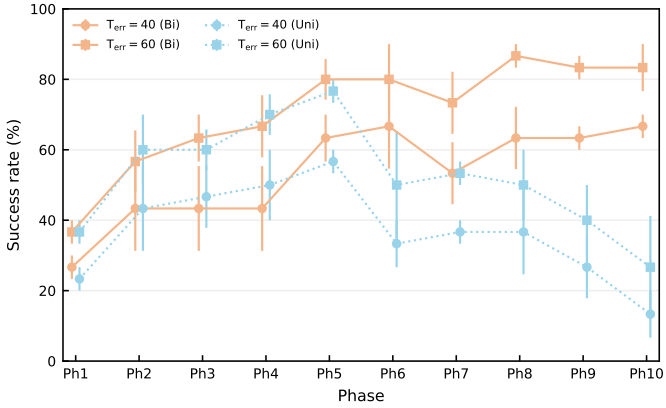

Fig. 10. **Success rate across developmental phases in the AI tutoring experiment.** Success rates are shown for the unidirectional (Uni) and bidirectional (Bi) tutoring conditions. Data points represent the mean success rate across training sets A-C, and error bars indicate the standard error of the mean (SEM). Two spatial error thresholds are used, T_{err}=40 and 60.

longer supports successful task execution. Additional examples of testing episodes are shown in Supplementary Video 3.

D. Qualitative and Quantitative Comparison in AI-Tutoring Experiment

We next examined qualitative and quantitative differences between the unidirectional and bidirectional AI-tutored trajectories, using the same analysis procedure as in the human-tutoring experiment. To provide an intuitive view of the AI-tutoring dynamics, Fig. 12 shows representative interaction episodes. In the unidirectional condition, the AI tutor generated the tutoring trajectory and imposed it on the robot. As shown at t=650, the robot-model panel shows a mismatch between the robot model's reconstruction and this imposed observation, whereas the AI-tutor-model panel shows alignment because the observation originated from the tutor model itself. This indicates that the tutor's guidance was externally imposed, rather than being shaped by the robot model's ongoing dynamics. In the bidirectional condition, by contrast, the observed trajectory was co-developed through interaction between the robot and the AI tutor. As shown at t=650, the robot-model panel shows that the co-developed observation remained aligned with the robot model's reconstruction, suggesting that the resulting trajectory became relatively consistent with the robot's ongoing dynamics. Importantly, this alignment did not result from the robot acting alone. During the reaching period toward the newly introduced object location (t=100–200), the increased intervention weight indicated that the AI tutor intervened when the robot model's prediction was not yet well adapted to the new object position. At the same time, the tutor-model panel shows that the tutor's reconstruction also remained aligned with the co-developed observation even when the intervention weight was near zero. This suggests that the tutor updated its internal state in accordance with the robot's ongoing behavior, allowing its guidance to remain closely coupled with the robot model's ongoing dynamics rather than being externally imposed. The corresponding video of this episode and additional examples are provided in Supplementary Video 4.

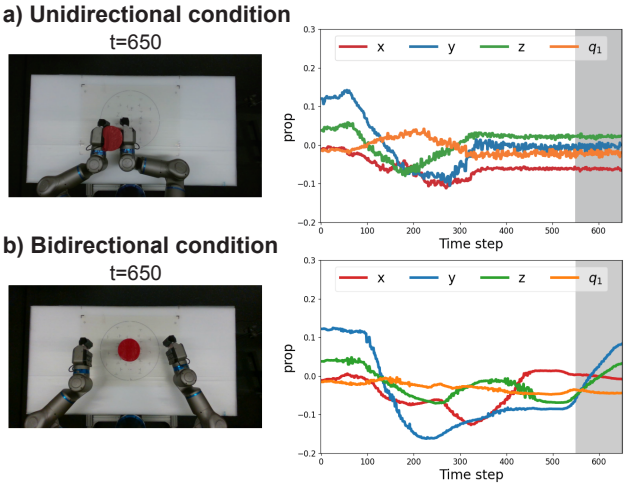

Fig. 11. **Representative testing episode in the AI-tutoring experiment at Phase 10.** Snapshots and corresponding proprioceptive trajectories are shown for the unidirectional and bidirectional conditions. The snapshots show the final state of the episode at t=650. The gray region indicates the regression window. In this representative episode, the unidirectional condition failed to complete the task: action generation remained near the same posture from around t=300 until the end of the episode, as reflected in the snapshot. In contrast, the bidirectional condition completed the task successfully, with the snapshot showing the robot returning toward the initial posture after placing the object at the designated return location.

Spatiotemporal Structure of AI-Tutored Trajectories: Fig. 13-a shows the left end-effector position and rotation trajectories obtained during AI tutoring in the unidirectional and bidirectional conditions. Consistent with the human-tutoring experiment, the unidirectional condition showed larger spatiotemporal variability across developmental phases, whereas the bidirectional condition produced more organized trajectories. In the bidirectional condition, task-critical variability remained around the reaching period, reflecting adjustment to newly introduced object locations, but overall trajectories were more aligned in shape and timing than in the unidirectional condition. Compared with the human-tutoring experiment, however, the bidirectional AI-tutored trajectories appeared to retain somewhat larger variability. This may reflect the more limited adaptability of the AI tutor, whose intervention was constrained by the learned tutor model and predefined intervention mechanism. Nevertheless, the key trajectory-level pattern also emerged under the AI-tutoring framework: bidirectional tutoring produced more organized trajectories than unidirectional tutoring while still allowing task-critical adjustments for newly introduced object locations.

Training Loss and Trajectory Compatibility in AI-Tutoring Experiment: We next examined the training loss, quantified as the variational free energy. As in the human-tutoring experiment, the loss value was interpreted as an index of how compatible the tutored trajectories were with the model's previously acquired sensorimotor dynamics.

As shown in Fig. 13-b, the bidirectional condition generally yielded lower training loss than the unidirectional condition across developmental phases and training sets. This difference remained significant in a linear model that accounted for developmental phase and training set (condition effect for bidirectional tutoring relative to unidirectional tutoring:

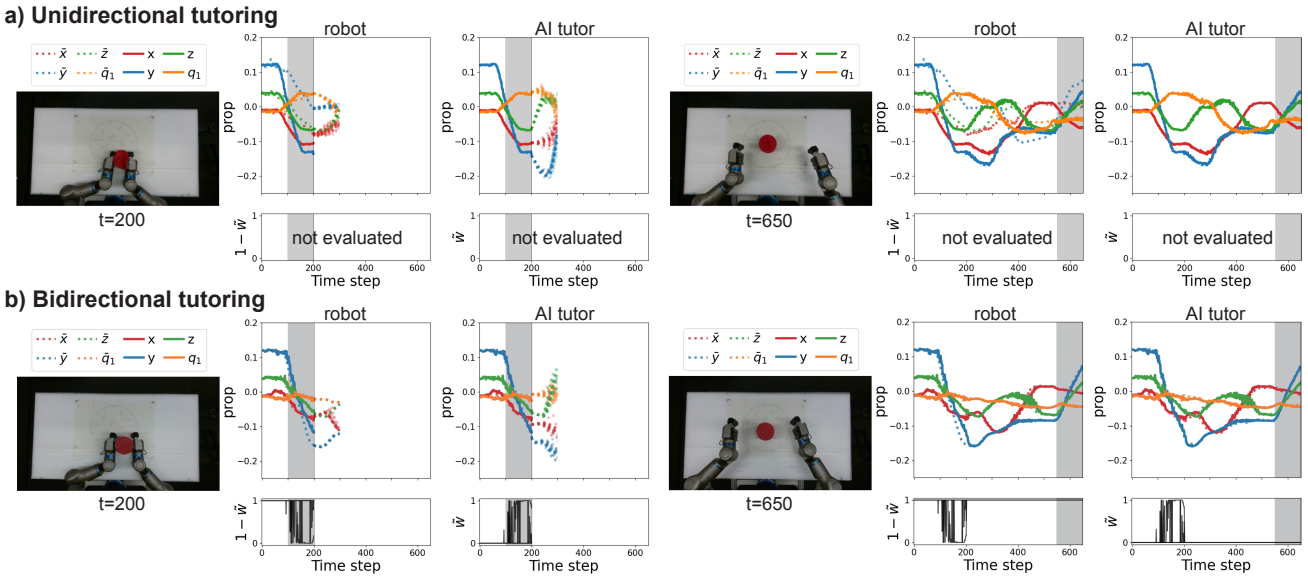

Fig. 12. **Representative examples of AI-tutoring dynamics at Phase 10. a** Unidirectional conditions. **b** Bidirectional conditions. The examples were selected from the same object-position set. Snapshots are shown at two representative time points: during reaching and grasping (t=200) and at the end of the episode (t=650). In each snapshot, the left and right panels show the trajectories evaluated from the robot model and the AI tutor model, respectively. Solid lines indicate the trajectory observed by each model, corresponding to the tutor-imposed trajectory in the unidirectional condition and the co-developed trajectory in the bidirectional condition. Dotted lines indicate each model's reconstructed trajectory before the current time point, and translucent dotted lines after the current time point indicate multiple future predictions generated from prior-sampled latent states. The intervention weight $\tilde{w}$ is shown only in the bidirectional condition and indicates the relative contribution of tutor intervention.

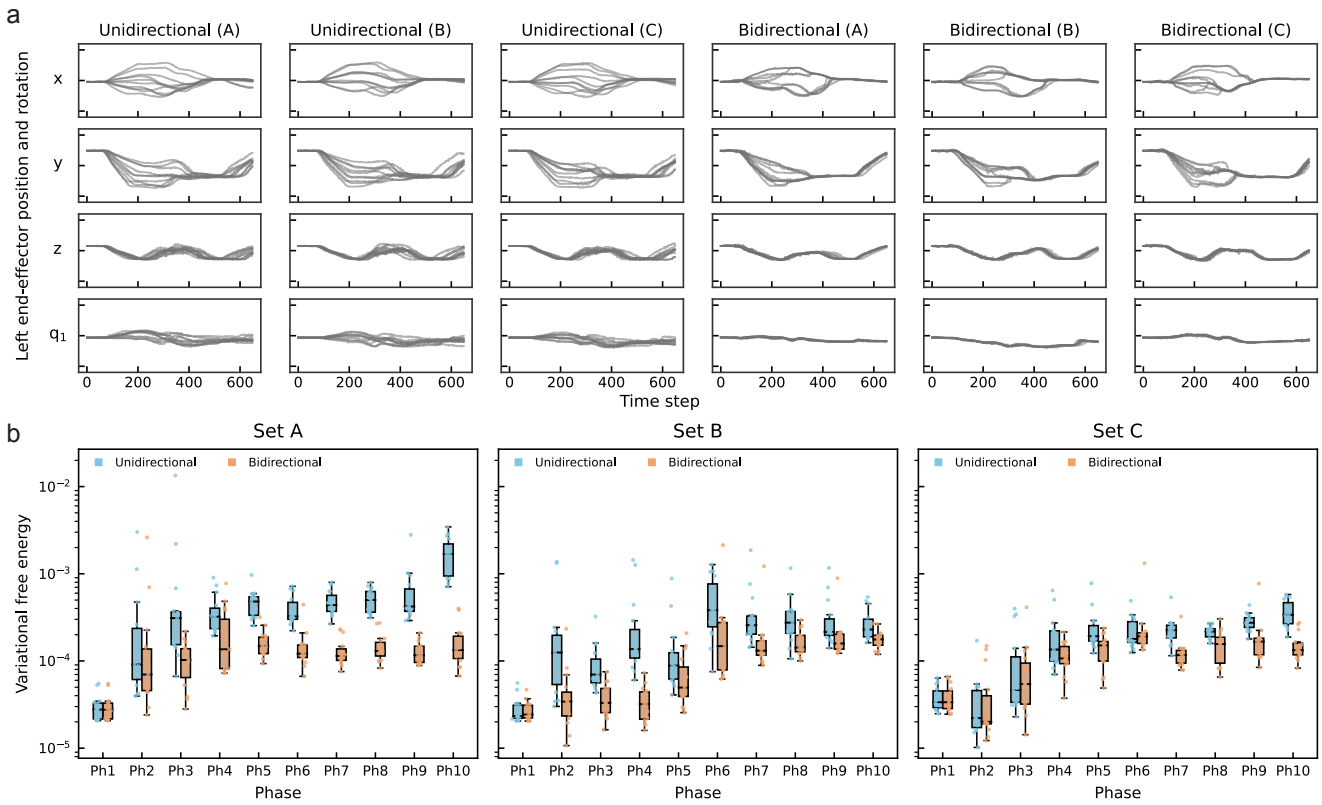

Fig. 13. **Qualitative and quantitative comparison of AI-tutoring dynamics under the unidirectional and bidirectional conditions. a** Left end-effector position and rotation trajectories obtained during AI-tutoring sessions, overlaid across the ten developmental phases for each training set (A–C). Each trace corresponds to the tutored trajectory at one phase. For rotation, only the first quaternion component (q_1) is shown, with the other components (q_2, q_3, and q_4) omitted for clarity. **b** Training loss, quantified as variational free energy, across developmental phases in the AI-tutoring experiment. Box plots show the distribution across random seeds. Statistical analysis was conducted on log-transformed values from Phases 2–10. Phase 1 was excluded from the statistical analysis because it served as a common baseline before the tutoring conditions diverged.

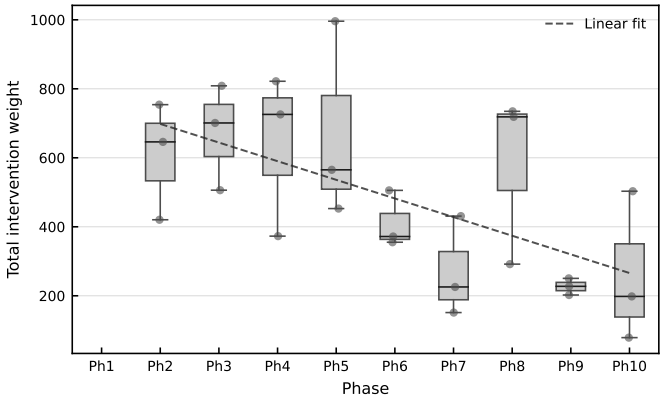

Fig. 14. **Changes in AI tutor intervention during bidirectional tutoring.** Box plots show the total intervention weight applied as tutor intervention during each tutoring episode in the bidirectional condition. Data from training sets A–C were pooled within each developmental phase. The dashed line indicates the linear trend estimated by linear regression across developmental phases. Total intervention weight decreased over phases (Pearson's $r{=}-0.781$, $p{=}0.0130$; linear regression $R^2{=}0.610$). No value is shown for Ph1 because this baseline phase did not include robot-to-tutor physical feedback, which was required to compute this measure.

$\beta{=}-0.805$, SE$=0.052$, $t{=}-15.50$, $p{<}0.001$). Thus, the quantitative training-loss pattern observed in the human-tutoring experiment was also found under the AI-tutoring framework. However, the separation between conditions was smaller than that observed in the human-tutoring experiment. This smaller difference may be related to the trajectory-level pattern shown in Fig. 13-a. Specifically, the bidirectional AI-tutored trajectories retained greater variability than those observed in the human-tutoring experiment, possibly reflecting the limited adaptability of the AI tutor compared with the human tutor. Nevertheless, the bidirectional condition still yielded lower training loss than the unidirectional condition, indicating that the same overall pattern was observed under the controlled AI-tutoring framework.

Reduction of AI Tutor Intervention across Developmental Phases: Finally, we examined the amount of AI tutor intervention in the bidirectional condition as an additional behavioral index of learning dynamics. Here, tutor intervention was quantified by the total intervention weight accumulated over each tutoring episode, $\sum_{t=1}^{T} \tilde{\boldsymbol{w}}_t$. As shown in Fig. 14, the total intervention weight showed an overall decreasing trend across developmental phases. As in the human-tutoring experiment, this reduction suggests that the robot gradually required less tutor intervention across developmental phases. Together with the increased task success rate observed in Fig. 10, this result indicates that the robot acquired task-relevant motor skills while becoming less dependent on AI tutor guidance.

VI. Discussion

In this study, we hypothesized that bidirectional tutoring supports stable developmental motor learning through the formation of co-developed trajectories that preserve behavioral coherence. The present results are consistent with this hypothesis across both human- and AI-tutoring experiments. In both settings, bidirectional tutoring resulted in higher and more stable task performance than unidirectional tutoring, particularly in later developmental phases (Fig. 5, Fig. 10).

Previous studies in robot learning from demonstration have emphasized that the quality and consistency of tutoring trajectories are crucial to forming stable sensorimotor representations and supporting generalization [22]–[24]. The present results support and extend this view by showing that such consistency can arise in a developmental tutoring process. In both human- and AI-tutoring experiments, bidirectional tutoring produced more spatiotemporally coherent trajectories than those in the unidirectional condition (Fig. 6-a, Fig. 13-a). These trajectories appeared to be consistent with the robot's previously acquired behavioral patterns. This suggests that, in bidirectional tutoring, the resulting tutoring experiences were naturally constrained by the robot's current generative dynamics, while tutor intervention introduced corrections for newly encountered task-relevant variations. Through this coupling, the co-developed trajectories maintained behavioral coherence across phases. The reduction of tutor intervention across developmental phases further supports the view that the coherent trajectories were co-developed rather than externally imposed by fixed tutor guidance (Fig. 7, Fig. 14). This indicates that the tutor did not simply ignore the robot's developing behavior, but adjusted its intervention according to the robot's improving motor skills.

This view is closely related to the broader idea that learning experiences are not fixed inputs, but are shaped by the learner's own behavior during ongoing interaction. For example, previous studies in reinforcement learning and imitation learning have discussed the influence of learner behavior on subsequent learning experiences in closed-loop interactions with the environment [47]–[49]. A related perspective is also found in sensorimotor contingency theory, which emphasizes the closed-loop coupling between action and sensory change [50]–[52]. The present study extends this closed-loop perspective to a tutoring context [4], i.e., socially grounded context in which learning experiences are co-developed through bidirectional interaction between the tutor and the learner. Specifically, bidirectional tutoring naturally incorporates the learner's internal dynamics, allowing them to guide the formation of consistent behavioral patterns.

Several limitations should be noted. First, the AI tutor reproduced only selected functional aspects of human tutoring, not the full richness of human adaptive behavior. However, it provided a controlled interaction setting under a fixed tutoring policy, which was sufficient for testing whether such bidirectional coupling could shape the resulting trajectories and support stable learning in the present study. Second, the present study was limited to a simple object manipulation task. In addition, the human-tutoring experiment involved a single tutor, although the procedure was repeated across three independently sampled object-position sets. These restrictions were partly due to the experimental cost of stage-wise developmental learning: even in this simple task, each experiment required repeated physical tutoring, model retraining, and evaluation across developmental phases. Thus, systematic comparisons involving multiple human participants, more complex tasks would require substantially more experimental time. Future

work should address both the limited adaptivity of the AI tutor and the experimental constraints of the present study by developing more autonomous and adaptive AI tutor agents. For example, large language models could support high-level interpretation of the robot's ongoing behavior and the generation of richer and more adaptive tutoring policies across developmental phases. Such agents could enable larger-scale and more systematic tutoring experiments, thereby reducing the burden of human-participant studies while allowing the resulting interaction principles to be validated later in real embodied human–robot interaction.

In summary, these findings highlight bidirectional tutoring as a co-developmental process in which new learning experiences are shaped jointly by tutor guidance and the learner's prior experience, rather than as the transfer of externally imposed demonstrations. This perspective may support the extension of bidirectional tutoring as an embodied and socially grounded approach to developmental motor learning in robots.

ACKNOWLEDGMENTS

J.T. was partially funded by the Japan Society for the Promotion of Science (JSPS) KAKENHI, Transformative Research Area (A): unified theory of prediction and action [26H01186].

APPENDIX

PV-RNN Forward Computation: Its computation of the hidden state $\boldsymbol{d}_t$ and prediction $\bar{\boldsymbol{x}}_t$ are as follows.

$$\boldsymbol{h}_t = (1-\frac{1}{\tau})\boldsymbol{h}_{t-1} + \frac{1}{\tau}(\boldsymbol{W}_{hd}\boldsymbol{d}_{t-1} + \boldsymbol{W}_{hz}\boldsymbol{z}_t + \boldsymbol{b}_h) \quad (10)$$

$$\boldsymbol{d}_t = \tanh(\boldsymbol{h}_t) \quad (11)$$

$$\bar{\boldsymbol{x}}_t = \tanh(\boldsymbol{W}_{\text{out}}\boldsymbol{d}_t + \boldsymbol{b}_{\text{out}}) \quad (12)$$

where $\boldsymbol{h}_t$ represents the hidden state before applying an activation function. τ is the time constant that determines how strongly past states contribute to the current state [53]. $\boldsymbol{W}_{hd}$, $\boldsymbol{W}_{hz}$, and $\boldsymbol{W}_{\text{out}}$ are weight matrices. $\boldsymbol{b}_h$ and $\boldsymbol{b}_{\text{out}}$ are bias terms. $\boldsymbol{z}_t$ is depended on prior $\boldsymbol{z}_t^p$ or posterior $\boldsymbol{z}_t^q$. $\boldsymbol{z}_t^p$ is represented as a distribution with mean $\boldsymbol{\mu}_t^p$ and variance $\boldsymbol{\sigma}_t^p$ as follows.

$$\boldsymbol{\mu}_t^p = \tanh\left(\boldsymbol{W}_{\mu}\boldsymbol{d}_{t-1} + \boldsymbol{b}_{\mu}\right) \quad (13)$$

$$\boldsymbol{\sigma}_t^p = \text{sigmoid}(\boldsymbol{W}_{\sigma}\boldsymbol{d}_{t-1} + \boldsymbol{b}_{\sigma}) \quad (14)$$

$$\boldsymbol{z}_t^p = \boldsymbol{\mu}_t^p + \boldsymbol{\sigma}_t^p * \boldsymbol{\epsilon}_t \quad (15)$$

where $\boldsymbol{\epsilon}_t$ is noise sampled from a standard normal distribution $\mathcal{N}(0,1)$ [54]. Posterior $\boldsymbol{z}_t^q$ is computed as follows

$$\boldsymbol{\mu}_t^q = \tanh\left(\boldsymbol{a}_t^{\mu}\right) \quad (16)$$

$$\boldsymbol{\sigma}_t^q = \text{sigmoid}(\boldsymbol{a}_t^{\sigma}) \quad (17)$$

$$\boldsymbol{z}_t^q = \boldsymbol{\mu}_t^q + \boldsymbol{\sigma}_t^q * \boldsymbol{\epsilon}_t \quad (18)$$

In our model, maximizing the ELBO corresponds to the maximizing expected log-likelihood of the observation given the model's prediction (accuracy term: e_t). Assuming a Gaussian likelihood with unit variance, the conditional probability can be expressed as

$$\ln p(\boldsymbol{x}_t|\boldsymbol{d}_t) \propto -\frac{1}{2}\left|\left|\boldsymbol{x}_t - \bar{\boldsymbol{x}}_t\right|\right|^2 \quad (19)$$

Algorithm 2 Error regression and action generation

for $t = 1$ to T **do**
  Initialize the adaptive vectors at time t:
  $$\boldsymbol{a}_t^{\mu} \leftarrow \text{arctanh}(\boldsymbol{\mu}_t^p), \quad \boldsymbol{a}_t^{\sigma} \leftarrow \text{logit}(\boldsymbol{\sigma}_t^p)$$
  Observe new targets $\boldsymbol{x}_t$
  Growing and shifting window
  $$t_{\text{start}} \leftarrow \max(1, t - t_{\text{win}})$$
  for $i = 1$ to N_{itr} **do**
    Generate $\bar{\boldsymbol{x}}_{t_{\text{start}}:t}$ with $\boldsymbol{a}_{t_{\text{start}}:t}$
    Compute $\mathcal{L}(\boldsymbol{\phi})$ and BPTT between t_{start} and t
    Update $\boldsymbol{a}_{t_{\text{start}}:t}$
  end for
  Generate $\bar{\boldsymbol{x}}_{t_{\text{start}}:t}$ with $\boldsymbol{a}_{t_{\text{start}}:t}$
  Generate prediction $\bar{\boldsymbol{x}}_{t+1}$ with sampling from the prior
  Send $\bar{\boldsymbol{x}}_{t+1,\text{prop}}$ to the controller for action generation
end for

and thus maximizing the log-likelihood is equivalent to minimizing the squared reconstruction error, defined as:

$$e_t = \frac{1}{2N_x}\sum_{i=1}^{N_x}||\boldsymbol{x}_{t,i} - \bar{\boldsymbol{x}}_{t,i}||^2 \quad (20)$$

where N_x is the dimension of $\boldsymbol{x}$. i represents the ith entry of each vector. Conversely, maximizing the ELBO also corresponds to minimizing the Kullback–Leibler divergence (KLD) between the approximate posterior and the prior (complexity term: r_t), defined as:

$$r_t = \frac{1}{N_z}D_{\text{KL}}[q_{\phi}(\boldsymbol{z}_t|\boldsymbol{d}_{t-1}, \boldsymbol{e}_{t:T})||p_{\theta}(\boldsymbol{z}_t|\boldsymbol{d}_{t-1})] \quad (21)$$

$$= \frac{1}{N_z}\sum_{i=1}^{N_z}\left\{\ln\frac{\boldsymbol{\sigma}_{t,i}^p}{\boldsymbol{\sigma}_{t,i}^q} + \frac{(\boldsymbol{\mu}_{t,i}^q - \boldsymbol{\mu}_{t,i}^p)^2 + (\boldsymbol{\sigma}_{t,i}^q)^2}{2(\boldsymbol{\sigma}_{t,i}^p)^2} - \frac{1}{2}\right\} \quad (22)$$

where N_z is the dimension of $\boldsymbol{z}$. i represents the ith entry of each vector.

Detailed Procedure for Online Inference and Action Generation: Algorithm 2 shows the detailed procedure for online inference and action generation through error regression with a growing and shifting window. As the time step advances, the adaptive vectors for the approximate posterior at the new time step are initialized from the corresponding prior using the inverse transformations of those used to compute the posterior parameters, namely $\text{arctanh}(\cdot)$ and $\text{logit}(\cdot)$ (see Appendix A, Forward Computation). The next-step prediction is then generated by sampling from the prior, and its proprioceptive component is provided to the controller for action generation.

REFERENCES

[1] A. N. Meltzoff and M. K. Moore, "Imitation of facial and manual gestures by human neonates," *Science*, vol. 198, no. 4312, pp. 75–78, 1977.

[2] J. M. PINE, "Tomasello, m., constructing a language: a usage-based theory of language acquisition. cambridge, ma: Harvard university press, 2003. pp. 388. hardback,£ 29.95. isbn 0-674-01030-2." *Journal of Child Language*, vol. 32, no. 3, pp. 697–702, 2005.

[3] H. De Jaegher and E. Di Paolo, "Participatory sense-making: An enactive approach to social cognition," *Phenomenology and the cognitive sciences*, vol. 6, no. 4, pp. 485–507, 2007.

[4] D. Wood, J. S. Bruner, and G. Ross, "The role of tutoring in problem solving," *Journal of child psychology and psychiatry*, vol. 17, no. 2, pp. 89–100, 1976.

[5] A. N. Meltzoff and W. Prinz, *The imitative mind: Development, evolution and brain bases*. Cambridge University Press, 2002, vol. 6.

[6] J. Lindblom and T. Ziemke, "Social situatedness of natural and artificial intelligence: Vygotsky and beyond," *Adaptive Behavior*, vol. 11, no. 2, pp. 79–96, 2003.

[7] L. S. Vygotsky, *Mind in society: The development of higher psychological processes*. Harvard university press, 1978, vol. 86.

[8] L. S. Vygotsky, *Thought and language*. MIT press, 2012, vol. 29.

[9] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini, "Developmental robotics: a survey," *Connection science*, vol. 15, no. 4, pp. 151–190, 2003.

[10] M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa, M. Ogino, and C. Yoshida, "Cognitive developmental robotics: A survey," *IEEE transactions on autonomous mental development*, vol. 1, no. 1, pp. 12–34, 2009.

[11] A. Cangelosi and M. Schlesinger, "From babies to robots: the contribution of developmental robotics to developmental psychology," *Child Development Perspectives*, vol. 12, no. 3, pp. 183–188, 2018.

[12] E. Ugur, Y. Nagai, E. Sahin, and E. Oztop, "Staged development of robot skills: Behavior formation, affordance learning and imitation with motionese," *IEEE Transactions on Autonomous Mental Development*, vol. 7, no. 2, pp. 119–139, 2015.

[13] S. Schaal, "Learning from demonstration," *Advances in neural information processing systems*, vol. 9, 1996.

[14] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[15] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[16] S. Haddadin and E. Croft, "Physical human–robot interaction," in *Springer handbook of robotics*. Springer, 2016, pp. 1835–1874.

[17] S. Murata, J. Namikawa, H. Arie, S. Sugano, and J. Tani, "Learning to reproduce fluctuating time series by inferring their time-dependent stochastic properties: Application in robot learning via tutoring," *IEEE Transactions on Autonomous Mental Development*, vol. 5, no. 4, pp. 298–310, 2013.

[18] W. Wang, Y. Chen, R. Li, and Y. Jia, "Learning and comfort in human–robot interaction: A review," *Applied Sciences*, vol. 9, no. 23, p. 5152, 2019.

[19] A. G. Billard, S. Calinon, and F. Guenter, "Discriminative and adaptive imitation in uni-manual and bi-manual tasks," *Robotics and Autonomous Systems*, vol. 54, no. 5, pp. 370–384, 2006.

[20] P. Kormushev, S. Calinon, and D. G. Caldwell, "Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input," *Advanced Robotics*, vol. 25, no. 5, pp. 581–603, 2011.

[21] M. Ito, K. Noda, Y. Hoshino, and J. Tani, "Dynamic and interactive generation of object handling behaviors by a small humanoid robot using a dynamic neural network model," *Neural Networks*, vol. 19, no. 3, pp. 323–337, 2006.

[22] J. Tani and M. Ito, "Self-organization of behavioral primitives as multiple attractor dynamics: A robot experiment," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 33, no. 4, pp. 481–488, 2003.

[23] Y. Wang, "Generalization capability for imitation learning," *arXiv preprint arXiv:2504.18538*, 2025.

[24] M. Sakr, J. Zhang, H. M. V. d. Loos, D. Kulić, and E. Croft, "Consistency matters: Defining demonstration data quality metrics in robot learning from demonstration," *ACM Transactions on Human-Robot Interaction*, vol. 15, no. 2, pp. 1–31, 2025.

[25] N. M. Shafiullah, Z. Cui, A. A. Altanzaya, and L. Pinto, "Behavior transformers: Cloning k modes with one stone," *Advances in neural information processing systems*, vol. 35, pp. 22 955–22 968, 2022.

[26] M. L. Latash, "The bliss (not the problem) of motor abundance (not redundancy)," *Experimental brain research*, vol. 217, no. 1, pp. 1–5, 2012.

[27] R. J. Van Beers, P. Haggard, and D. M. Wolpert, "The role of execution noise in movement variability," *Journal of neurophysiology*, vol. 91, no. 2, pp. 1050–1063, 2004.

[28] H. G. Wu, Y. R. Miyamoto, L. N. G. Castro, B. P. Ölveczky, and M. A. Smith, "Temporal structure of motor variability is dynamically regulated and predicts motor learning ability," *Nature neuroscience*, vol. 17, no. 2, pp. 312–321, 2014.

[29] K. Friston, "A theory of cortical responses," *Philosophical transactions of the Royal Society B: Biological sciences*, vol. 360, no. 1456, pp. 815–836, 2005.

[30] A. Clark, *Surfing uncertainty: Prediction, action, and the embodied mind*. Oxford University Press, 2015.

[31] J. Hohwy, *The predictive mind*. OUP Oxford, 2013.

[32] A. Ahmadi and J. Tani, "A novel predictive-coding-inspired variational rnn model for online prediction and recognition," *Neural computation*, vol. 31, no. 11, pp. 2025–2074, 2019.

[33] J. Borojevic, G. W. Haddon-Hill, J. Sandoval, and S. Murata, "Deep active inference in physical human-robot interaction: Balancing exploration and goal-directed behavior," in *2026 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2026, pp. 1504–1509.

[34] J. Tani, R. Nishimoto, J. Namikawa, and M. Ito, "Codevelopmental learning between human and humanoid robot using a dynamic neural-network model," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 1, pp. 43–59, 2008.

[35] B. Akgun, M. Cakmak, J. W. Yoo, and A. L. Thomaz, "Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective," in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, 2012, pp. 391–398.

[36] S. Ikemoto, H. B. Amor, T. Minato, B. Jung, and H. Ishiguro, "Physical human-robot interaction: Mutual learning and adaptation," *IEEE robotics & automation magazine*, vol. 19, no. 4, pp. 24–35, 2012.

[37] T. Matsumoto, W. Ohata, and J. Tani, "Incremental learning of goal-directed actions in a dynamic environment by a robot using active inference," *Entropy*, vol. 25, no. 11, p. 1506, 2023.

[38] H. Sawada, W. Ohata, and J. Tani, "Human–robot kinaesthetic interactions based on the free-energy principle," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2024.

[39] S. Amari, "A theory of adaptive pattern classifiers," *IEEE Transactions on Electronic Computers*, no. 3, pp. 299–307, 2006.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[41] W. Ohata and J. Tani, "Characterizing the sense of agency in human–robot interaction based on the free energy principle," *npj Complexity*, vol. 2, no. 1, p. 12, 2025.

[42] J. Tani, "An interpretation of the "self" from the dynamical systems perspective: A constructivist approach," *Journal of Consciousness Studies*, vol. 5, no. 5–6, pp. 516–542, 1998.

[43] R. French, "Catastrophic forgetting in connectionist networks: Causes, consequences and solutions," *Trends in Cognitive Sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[44] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," *Advances in neural information processing systems*, vol. 30, 2017.

[45] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*. Elsevier, 1989, vol. 24, pp. 109–165.

[46] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in cognitive sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[47] R. S. Sutton, A. G. Barto *et al.*, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[48] Y. Liu, A. Swaminathan, A. Agarwal, and E. Brunskill, "Off-policy policy gradient with state distribution correction," *arXiv preprint arXiv:1904.08473*, 2019.

[49] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[50] J. K. O'regan and A. Noë, "A sensorimotor account of vision and visual consciousness," *Behavioral and brain sciences*, vol. 24, no. 5, pp. 939–973, 2001.

[51] A. Lübbert, F. Göschl, H. Krause, T. R. Schneider, A. Maye, and A. K. Engel, "Socializing sensorimotor contingencies," *Frontiers in Human Neuroscience*, vol. 15, p. 624610, 2021.

[52] L. Jacquey, G. Baldassarre, V. G. Santucci, and J. K. O'regan, "Sensorimotor contingencies as a key drive of development: from babies to robots," *Frontiers in neurorobotics*, vol. 13, p. 98, 2019.

[53] R. D. Beer, "On the dynamics of small continuous-time recurrent neural networks," *Adaptive Behavior*, vol. 3, no. 4, pp. 469–509, 1995.

[54] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

SUPPLEMENTARY MATERIAL

A. Generation of AI-Tutor Training Trajectories

The AI-tutor training trajectories were generated computationally rather than recorded from human demonstrations. The purpose of this procedure was to provide task-successful trajectories that preserved the overall manipulation sequence while including spatiotemporal variability in detailed movement execution. The reference object-grasping postures of both end effectors were defined based on a successful center-object grasping configuration. Each generated trajectory followed the same task-level sequence of reaching, grasping, lifting, placing, releasing, and returning. The sequence was specified by task-relevant waypoints,

$$\mathcal{V} = \{v_{\text{home}}, v_{\text{reach}}, v_{\text{grasp}}, v_{\text{lift}}, v_{\text{above center}}, v_{\text{place center}}, v_{\text{return}}\}.$$

which corresponded to the object-reaching position, lift-up position, position above the center of the workspace, placing position at the center of the workspace, and return position near the arm home posture, respectively. Here, v_{grasp} was defined by shifting the left and right end-effector components of v_{reach} by 1 cm toward the object center along the lateral direction, so that the hands moved inward to grasp the object.

Variability was introduced by modifying the task-relevant waypoints in $\mathcal{V}$ in a structured manner. First, object-location variability was introduced by translating the reaching waypoint v_{reach} according to one of 37 object-location conditions defined on a grid over the workspace. The same 2D displacement was applied to the left and right end-effector components of v_{reach}, thereby changing the target object location while preserving the relative bimanual grasping configuration. Second, movement-shape variability was introduced by perturbing selected non-critical waypoints in $\mathcal{V}$. Specifically, random Cartesian perturbations were applied to v_{lift}, $v_{\text{above center}}$, and v_{return}, while $v_{\text{place center}}$ was kept fixed to preserve the designated return location. For v_{lift} and $v_{\text{above center}}$, the same Cartesian perturbation was applied to both end effectors, sampled uniformly within ± 5 cm along each Cartesian axis. For v_{return}, independent Cartesian perturbations were applied to the left and right end effectors, also sampled uniformly within ± 5 cm. This allowed the generated trajectories to vary in detailed movement shape while preserving task-critical reaching and placing constraints. Finally, rotational variability was introduced by perturbing the end-effector orientations associated with the selected waypoints. Random Euler-angle perturbations were sampled uniformly within $\pm 5^\circ$ around the x and y axes and $\pm 10^\circ$ around the z axis, and the resulting orientations were converted to quaternion targets.

Temporal variability was introduced by randomly perturbing the duration of each movement segment. The base segment durations are shown in Table I. For each generated sequence, the duration of each segment except the final one was randomly jittered within approximately one third of its base duration, and the final segment duration was adjusted so that the total sequence length remained fixed at $T{=}650$. This changed the timing of reaching, grasping, lifting, placing, and returning while preserving the overall task sequence

TABLE I
BASE SEGMENT DURATIONS.

Segment (waypoint relation)	Steps
Home (stay at v_{home})	75
Reach ($v_{\text{home}} \rightarrow v_{\text{reach}}$)	75
Grasp ($v_{\text{reach}} \rightarrow v_{\text{grasp}}$)	10
Stay after grasping (stay at v_{grasp})	30
Lift ($v_{\text{grasp}} \rightarrow v_{\text{lift}}$)	45
Stay after lifting (stay at v_{lift})	30
Move above ($v_{\text{lift}} \rightarrow v_{\text{above}}$)	45
Stay above (stay at v_{above})	30
Put down ($v_{\text{above}} \rightarrow v_{\text{place}}$)	45
Stay after placing (stay at v_{place})	80
Release (open hands at v_{place})	10
Return ($v_{\text{place}} \rightarrow v_{\text{return}}$)	75
Final stay (stay at v_{return})	100

After the waypoints and segment durations were determined, each trajectory was generated by interpolating between successive waypoints for both position and orientation. The generation procedure was repeated for 37 object-location conditions and 15 random seeds, resulting in 555 AI-tutor training trajectories. Because the same task-level waypoint structure was used across all trajectories, the generated sequences preserved the overall manipulation sequence. At the same time, they differed in object location, movement timing, movement shape, and end-effector orientation. Thus, the AI tutor was trained on variable task-successful executions rather than repeated deterministic templates.

B. Supplementary Videos

- Supplementary Video 1 shows representative autonomous testing episodes in the human-tutoring experiment.
- Supplementary Video 2 shows representative human-tutoring episodes.
- Supplementary Video 3 shows representative autonomous testing episodes in the AI-tutoring experiment.
- Supplementary Video 4 shows representative AI-tutoring episodes.

C. Data and Code Availability

The processed data and analysis code used in this study are available at: https://github.com/oist-cnru/tcds-motor-skill-tutoring-code-data.