2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

CATALYST: Cognitive-to-Autonomy-inspired Two-Stage Training Data Generation with Local-System-Aware Selection Technique

Taehoon Kim and Sehoon Oh

***Abstract*— In conventional learning-based robotic dynamics modeling, physical information is mostly incorporated into the model or loss function, while the design of training data often relies on random sampling or uniform coverage, which can limit performance. To address this gap, this paper proposes the Cognitive-to-Autonomy-inspired Two-stage trAining data generation with Local-sYstem-aware Selection Technique (CATALYST) framework, which generates optimal training data based on physics priors and the modeling structure of the chosen learning model. Stage 1 uses the CAD-derived inertia matrix $M(q)$ to approximate the joint distribution of $[q, M]$ with a Probabilistic Local Model (PLM), thereby identifying the optimal locations for the local model centers (μ_k^{opt}). Stage 2 then optimizes an Operating-Point-Centered Excitation Trajectory (OPCET). This optimization simultaneously (i) aligns the trajectory with the target operating points (l_m), (ii) enforces range-of-motion (RoM) constraints (l_r), and (iii) achieves desirable velocity–acceleration statistics (large volume, isotropy, low correlation, captured by l_s). We validate the approach in simulation using a 3-DoF yaw–pitch–pitch manipulator, which allows visual demonstration of the process and outcomes. We then analyze the framework step by step. Results show that each stage meets its objective. A PLM trained on data generated by the proposed trajectories outperforms baselines (Spread/RoM, ill-centered, Tukey-windowed chirp, and cubic) in both torque regression and control. Thus, CATALYST yields more accurate regression and more reliable feedforward control than conventional designs.**

I. Introduction

In modern robotics research, two dominant approaches to control-oriented modeling have emerged: model-based approaches, which rely on explicit analytical formulations of robot dynamics, and data-driven approaches, which directly exploit collected data. Recently, hybrid frameworks that integrate the two paradigms have gained increasing attention. For example, physical consistency and Lagrangian-based structures have been incorporated into the loss functions or architectures of learning models, thereby improving generalization and sample efficiency in data-driven dynamics learning [1], [2]. However, most of these efforts inject physical information only into the model or the loss function, while the generation of training data itself is still largely based on random sampling or naive coverage designs. In robotics, physical laws are often embedded in models or losses but rarely in training-data design. Random data collection can bias data quality and degrade performance; physics-guided data generation is therefore needed.

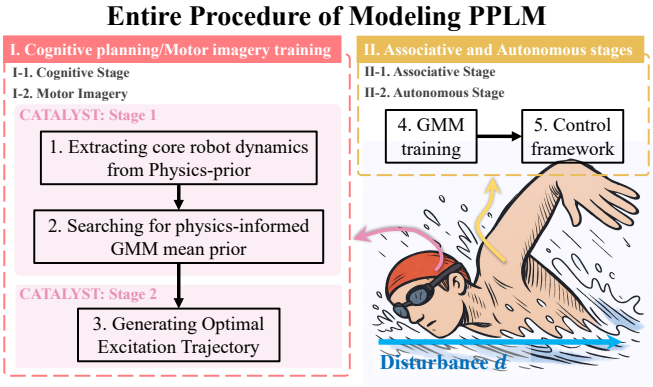

Fig. 1. A conceptual diagram illustrating the modeling procedure of Physics-inspired PLM (PPLM). The overall process is inspired by the human motor skill acquisition model. In this paper, the CATALYST framework is proposed as an algorithmic representation of the cognitive stage and motor imagery, adapted for PLM training.

Recent work has pursued active exploration and simulator refinement, where informative data are collected and small amounts of real-world data calibrate simulators, improving transfer in identification and control [3], [4]. However, these studies offer few physics-based criteria for where and how to collect data, and rarely target data that are optimal for the learning model.

Parametric and non-parametric approaches are two primary methods used to model robot dynamics. Parametric models describe the robot system using parameters like inertia, mass, center of mass, and link lengths [5]. Traditional system identification techniques, such as least squares [6], frequency response analysis [7], and optimization [8], have been employed to model and obtain parameters for complex robot dynamics. While efficient and simple, they have limitations, especially in capturing nonlinear behaviors and interactions among components, leading to discrepancies between simulations and actual implementations [9].

To address the limitations of parametric approaches, data-driven methods have been actively studied [10], [11]. However, these methods have been reported to suffer from poor generalization performance and training inefficiency [12], [13], which often stem not from limitations in model structure, but from the lack of deliberate and informative data generation. Even in parametric approaches, persistent

*This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2025-02219277; AI Star Fellowship Support (DGIST)), the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00354028), and the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. RS-2025-25420118).

Taehoon Kim and Sehoon Oh are with the Department of Robotics and Mechatronics Engineering, DGIST, Daegu 42988, Republic of Korea (sean9595@dgist.ac.kr; corresponding author: sehoon@dgist.ac.kr).

excitation and careful experiment design have been shown to be essential for obtaining reliable models, with Fisher information–based trajectory optimization serving as a representative example [14], [15]. Recently, there have also been reports of improved identification accuracy and sim-to-real performance through active exploration and simulator calibration using small amounts of real-world data [3]. Additionally, DAgger has demonstrated in the field of imitation learning that collecting on-policy data is critical for generalization [16]. Ultimately, random data collection is a major cause of degraded learning performance, and research on designing training data that leverages both physical prior knowledge and the learning model's structure remains rare in dynamics modeling research.

To address these challenges, we present a framework that integrates learning-based modeling with analytical rigid-body dynamics for manipulator modeling. The learning component is a *Probabilistic Local Model (PLM)* instantiated with Gaussian Mixture Models and Gaussian Mixture Regression (GMM/GMR), which partition the input space into interpretable local regions and provide closed-form predictions suitable for real-time control; the probabilistic formulation further captures uncertainty and multimodality for robust estimation–control integration [17], [18]. Central to our approach is *the physics-guided design of training references, so that the learned PLM becomes a physics-inspired PLM (PPLM) rather than a purely data-driven model*. We introduce CATALYST, a two-stage data-generation scheme: Stage 1 selects physically informative local centers using the functional dependence of the inertia matrix on joint angles and Stage 2 synthesizes physically constrained trajectories that drive sufficient visitation of those centers. Complete formulations are provided in Section III.

II. Preliminaries and Limitations

A. Advantages of PLM in Dynamics Modeling

A PLM with GMM/GMR represents robot dynamics as a weighted combination of local linear regressors, where a gating function selects and weights the local models given the state. This structure yields closed-form predictions and makes inference complexity independent of the number of training samples, enabling real-time use in control loops. Moreover, component covariances encode uncertainty and correlations, which is beneficial for robust estimation–control integration. A mixture-of-experts interpretation further clarifies how operating-point-centric modeling arises naturally. The regression equations used in this work are summarized in Sec. II.B (Eqs. (1)–(3)).

B. PLM design scheme for robot manipulator dynamics modeling

PLM approximates the joint distribution of $[x, y]^\top$, where $x = [q, \dot{q}, \ddot{q}]^\top$ denotes the input (joint states) and $y = \tau$ denotes the output (joint torques), as a mixture of multivariate Gaussians:

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim \sum_{k=1}^{C} \pi_k \mathcal{N}\left(\begin{bmatrix} \mu_k^x \\ \mu_k^y \end{bmatrix}, \begin{bmatrix} \Sigma_k^{xx} & \Sigma_k^{xy} \\ \Sigma_k^{yx} & \Sigma_k^{yy} \end{bmatrix} \right), \tag{1}$$

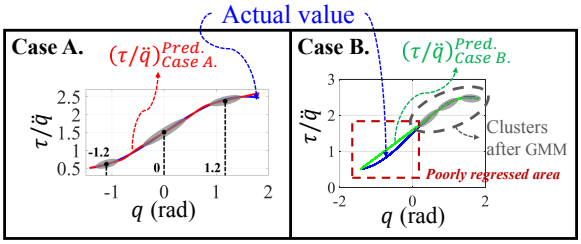

Fig. 2. Importance of cluster positioning for efficient PLM modeling. In Case A, clusters are properly positioned, while in Case B, their arrangement is not carefully considered during data extraction. Gray ellipses represent clusters for $[q, \tau/\ddot{q}]^\top$, with the center of each ellipse indicating the cluster mean.

where C is the number of components, π_k are the mixture coefficients, and $\mu_k^{(\cdot)}$, $\Sigma_k^{(\cdot)(\cdot)}$ denote the mean and covariance of each Gaussian. After parameter estimation with the Expectation-Maximization (EM) method, regression is performed by conditioning on x:

$$\hat{y}(x) = \sum_{k=1}^{C} h_k(x)\, \tilde{y}_k(x), \tag{2}$$

$$\begin{aligned} \tilde{y}_k(x) &= \mu_k^y + \Sigma_k^{yx} (\Sigma_k^{xx})^{-1}(x - \mu_k^x), \\ h_k(x) &= \frac{\pi_k \mathcal{N}(x \mid \mu_k^x, \Sigma_k^{xx})}{\sum_{i=1}^{C} \pi_i \mathcal{N}(x \mid \mu_i^x, \Sigma_i^{xx})}. \end{aligned} \tag{3}$$

Each component regression $\hat{y}_k(x)$ is a local linear model, and the overall prediction is interpreted as their weighted average [19]. Rearranging (2) yields:

$$\begin{aligned} \tilde{y}_k(x) &= \mu_k^y + \Sigma_k^{yx}(\Sigma_k^{xx})^{-1}(x - \mu_k^x) \\ &= \underbrace{\Sigma_k^{yx}(\Sigma_k^{xx})^{-1}}_{A_k} \cdot x + \underbrace{\mu_k^y - \Sigma_k^{yx}(\Sigma_k^{xx})^{-1}\mu_k^x}_{B_k}, \\ \therefore \hat{y}(x) &= \sum_{k=1}^{C} h_k(x)\, \tilde{y}_k(x), \\ &= \sum_{k=1}^{C} h_k(x) \cdot (A_k x + B_k). \end{aligned} \tag{4}$$

In the above equations, each local linear model $A_k x + B_k$ is weighted by its gating function $h_k(x)$. The weight $h_k(x)$ is essentially the conditional probability (from Eq. (3)) that determines the contribution of the k-th local model given input x. Thus, GMR forms a probabilistic local model: the gate $h_k(x)$ weights local linear regressors $A_k x + B_k$.

Consequently, regression performance strongly depends on how well the cluster means and covariance structures capture dynamically critical variations. The proposed method in this paper directly addresses this issue by introducing physics-guided strategies for positioning these clusters in meaningful regions.

C. Limitations of PLM in Robot Dynamics

Despite these advantages, PLM also faces fundamental challenges when applied to robot dynamics. Since the EM

method tends to allocate clusters to regions with high data density [20], [21], training data that does not adequately represent dynamically important regions can lead to poor approximation of key nonlinearities. This often results in unnecessary increases in the number of components, higher computational complexity, and degraded generalization performance.

As an illustrative example in Fig. 2, consider a simple single-DoF (Degree of Freedom) system where inertia depends on the joint angle:

$$M(q)\,\ddot{q} = (\sin q + 1.5)\,\ddot{q} = \tau. \tag{5}$$

Here, $M(q) = \tau/\ddot{q}$ changes significantly depending on both the sign and magnitude of q. Effective modeling requires that clusters be strategically placed around regions where such changes are pronounced. However, naive coverage or uniformly sampled datasets may oversample regions of low dynamic variation, causing clusters to miss critical behaviors.

Therefore, designing training data such that cluster centers converge to dynamically meaningful operating points is essential. This motivates the two-step data design framework proposed in this paper, where physical insights guide both cluster placement and trajectory generation, bridging the gap between the inherent strengths of PLM and the requirements for accurate robot dynamics modeling.

III. CATALYST Framework: Training Data Generation with Physics-based Priors for PPLM

As noted in Sec. II.A, local models of PLM can capture configuration-dependent variability, yet EM tends to allocate components by data density rather than dynamical importance. Without deliberate data design, inertia-varying or strongly coupled regions may be underrepresented.

Related work [22] attempted to incorporate these dynamics into probabilistic local modeling, but it faced notable limitations. First, the means of local models were selected heuristically (often visually). For high-DoF manipulators, this manual process is impractical, and even when candidate points are chosen, there is no guarantee that they are truly optimal from the perspective of GMM learning. To address this issue, the present work introduces an optimization-based algorithm that leverages physics-based priors to determine optimal mean points in a principled manner. Second, the trajectory generation in earlier work was designed such that samples converged toward the selected mean points using a simple ℓ_2 cost. However, this formulation did not guarantee that the resulting GMM cluster means would actually align with those locations, and it also ignored correlations among q, $\dot{q}$, and $\ddot{q}$ in the cost function. These limitations motivate the need for a more systematic and physically grounded framework.

We address this with **C**ognitive-to-**A**utonomy-inspired **T**wo-stage tr**A**ining data generation with **L**ocal-s**Y**stem-aware **S**election **T**echnique (CATALYST) framework, a two-stage, physics-guided data-generation scheme: **Stage 1** places informative local-model centers $\{\mu_k^{\mathrm{opt}}\}$ using CAD-derived inertia $M(q)$, **Stage 2** constructs an Operating-Point-Centered Excitation Trajectory (OPCET) that visits those centers while satisfying RoM/rate limits and promoting large, isotropic, low-correlation velocity–acceleration statistics. This embeds physics in the data as well as in the model, yielding a physics-inspired PLM (PPLM) (see Fig. 1).

Algorithm 1: Optimal Mean Point Searching

Result: Optimal cluster count $C^\star$ and mean points $\boldsymbol{\mu}^{opt} := \{\boldsymbol{\mu}_k^{opt}\}_{k=1}^{C^\star}$

Require: Number of sample group N, $\boldsymbol{q}$ with $\boldsymbol{q}_{min}$ and $\boldsymbol{q}_{max}$, $M^{CAD}(\boldsymbol{q})$, candidate range $[C_{\min}, C_{\max}]$, tolerance l_{tol}

0. Precompute:

1) Sample q within the RoM.

2) Calculate inertia matrix $M^{CAD}(\boldsymbol{q})$ based on given CAD data (A URDF-generated dataset $[\boldsymbol{q}, M^{CAD}]^\top$ replaces the analytical inertia matrix expression).

for $C \leftarrow C_{\min}$ **to** $C_{\max}$ **do**

1. Initialization:

$\boldsymbol{\psi}_k = [\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k]$ for each cluster $(k = 1, \ldots, C)$;

2. Find optimal $\boldsymbol{\psi}$ for this C:

while $\Delta e > l_{tol}$ **do**

$$\min_{\boldsymbol{\psi}} \quad e = \sum_{i=1}^{N} \Big(M^{pred}(\boldsymbol{q}_i) - M^{CAD}(\boldsymbol{q}_i) \Big)^2$$

where

$$h_k(\boldsymbol{q}_i) = \frac{\pi_k \cdot \mathcal{G}(\boldsymbol{q}_i | \boldsymbol{\mu}_k^{\boldsymbol{q}}, \boldsymbol{\Sigma}_k^{\boldsymbol{q}})}{\sum_{j=1}^{C} \pi_j \cdot \mathcal{G}(\boldsymbol{q}_i | \boldsymbol{\mu}_j^{\boldsymbol{q}}, \boldsymbol{\Sigma}_j^{\boldsymbol{q}})},$$

$$\hat{\boldsymbol{y}}_k(\boldsymbol{q}_i) = \boldsymbol{\mu}_k^{\boldsymbol{y}} + \boldsymbol{\Sigma}_k^{yq} (\boldsymbol{\Sigma}_k^{qq})^{-1} (\boldsymbol{q}_i - \boldsymbol{\mu}_k^{q}),$$

$$M^{pred}(\boldsymbol{q}_i) = \sum_{k=1}^{C} h_k(\boldsymbol{q}_i) \cdot \hat{\boldsymbol{y}}_k(\boldsymbol{q}_i).$$

end

Compute $\mathrm{AIC}(C) = 2N_{\mathrm{param}} - 2\log L$

if $\mathrm{AIC}(C) < \mathrm{AIC}_{best}$ **then**

$\mathrm{AIC}_{best} \leftarrow \mathrm{AIC}(C)$;;

$C^\star \leftarrow C$;;

$\boldsymbol{\psi}^{opt} \leftarrow \boldsymbol{\psi}$;

end

end

Return: $\boldsymbol{\mu}^{opt} := \{\boldsymbol{\mu}_k\}_{k=1}^{C^\star}$ extracted from $\boldsymbol{\psi}^{opt}$

1) Stage 1: Search for Optimal Local-Model Means: The ultimate goal of this study is to build a dynamic model that accurately predicts the robot's torque τ. Assuming that nonlinear terms (coriolis/centrifugal, gravitational and friction torque) are either neglected or compensated for, the effective dynamics can be expressed as

$$\begin{aligned} &M(q)\ddot{q} + N(q, \dot{q}, \ddot{q}) = \tau, \\ &\text{where} \quad N(q, \dot{q}, \ddot{q}) = -(\Delta C + \Delta g) + n(q, \dot{q}, \ddot{q}). \end{aligned} \tag{6}$$

Algorithm 2: OPCET generation

Input: Sampling time T_s, horizon T_{fin}, number of sinusoids N_{sinu}, design frequencies $\boldsymbol{\omega}_{\text{des}}$, target means $\{\boldsymbol{\mu}_k^{\text{opt}}\}_{k=1}^{C}$, bounds $(\boldsymbol{q}_{\min}, \boldsymbol{q}_{\max}, \dot{\boldsymbol{q}}_{\min}, \dot{\boldsymbol{q}}_{\max}, \ddot{\boldsymbol{q}}_{\min}, \ddot{\boldsymbol{q}}_{\max})$, weights $(\lambda_\mu, \lambda_{\ell_2}, \lambda_{\text{nll}}, \lambda_{\text{iso}}, \lambda_{\text{decol}}, w_s, w_r)$, regularizers ϵ, ε, fixed mixture params $\{\pi_k, \Sigma_k\}_{k=1}^{C}$.

Output: Joint trajectory $\boldsymbol{q} \in \mathbb{R}^{N \times n_q}$.

1. Initialization: Set trajectory parameters per joint i as $\boldsymbol{\phi}_i = [\boldsymbol{a}_i, \boldsymbol{b}_i, q_{0,i}]$.

2. Generate trajectory for each joint: for each i,

$$q_i(\boldsymbol{\phi}_i, t) = \sum_{j=1}^{N_{\text{sinu}}} \Big(a_{ij} \sin(\omega_{\text{des},ij} t) + b_{ij} \cos(\omega_{\text{des},ij} t) \Big) + q_{0,i}.$$

while $\Delta e > l_{tol}$ **do**

$$\min_{\boldsymbol{\phi}} \quad l_m(\boldsymbol{\phi}) + w_r\, l_r(\boldsymbol{\phi}) + w_s\, l_s(\boldsymbol{\phi})$$

$$\text{s.t.} \quad \dot{\boldsymbol{q}}_{\min} \le \dot{\boldsymbol{q}}(\boldsymbol{\phi}) \le \dot{\boldsymbol{q}}_{\max}, \quad \ddot{\boldsymbol{q}}_{\min} \le \ddot{\boldsymbol{q}}(\boldsymbol{\phi}) \le \ddot{\boldsymbol{q}}_{\max},$$
$$\boldsymbol{q}_{\text{init}} = \boldsymbol{0},\ \boldsymbol{q}_{\text{fin}} = \boldsymbol{0},$$
$$\dot{\boldsymbol{q}}_{\text{init}} = \boldsymbol{0},\ \dot{\boldsymbol{q}}_{\text{fin}} = \boldsymbol{0},$$
$$\ddot{\boldsymbol{q}}_{\text{init}} = \boldsymbol{0},\ \ddot{\boldsymbol{q}}_{\text{fin}} = \boldsymbol{0}.$$

Define losses:

$$l_m(\boldsymbol{\phi}) := \underbrace{\lambda_\mu l_\mu(\boldsymbol{\phi})}_{\text{GMM mean}} + \underbrace{\lambda_{\ell_2} l_{\ell_2}(\boldsymbol{\phi})}_{\ell_2} + \underbrace{\lambda_{\text{nll}} l_{\text{nll}}(\boldsymbol{\phi})}_{\text{neg. log-likelihood}},$$

$$l_r(\boldsymbol{\phi}) := \big(\max\, \boldsymbol{q}(\boldsymbol{\phi}) - \boldsymbol{q}_{\max} \big)^2 + \big(\min\, \boldsymbol{q}(\boldsymbol{\phi}) - \boldsymbol{q}_{\min} \big)^2,$$

$$l_s(\boldsymbol{\phi}) := \underbrace{l_{\text{vol}}(\boldsymbol{\phi})}_{\text{spreading}} + \underbrace{\lambda_{\text{iso}} l_{\text{iso}}(\boldsymbol{\phi})}_{\text{isotropy}} + \underbrace{\lambda_{\text{decol}} l_{\text{decol}}(\boldsymbol{\phi})}_{\text{decorrelation}},$$

end

Here, $\Delta C = \hat{C} - C$ and $\Delta g = \hat{g} - g$ represent modeling errors. Stage 1 focuses on the inertial term $M(q)\ddot{q}$, which primarily influences large variations in τ. The goal is to find GMM local-model centers that capture salient variation in $M(q)$, which depends on the position q. Accordingly, we search for a set of optimal means μ_k^{opt} that serve as physically meaningful operating points for the local models. The rationale is as follows. First, in high-acceleration regions, τ is often dominated by $M(q)\ddot{q}$, so placing local models where $M(q)$ can be well-represented is advantageous in terms of regression error and generalization. Second, since $M(q)$ depends only on the joint angles q and can be precomputed from the robot's CAD model, it provides reliable physical information that is unaffected by uncertainties such as sensor noise or data collection bias. As a result, if μ^{opt} is aligned with physically significant points such as regions of rapid variation or coupling in $M(q)$, one can expect lower prediction error and higher generalization performance even with the same data budget.

To achieve this, joint positions q are sampled randomly but well-distributed across the entire RoM to create a virtual dataset $\{q_i\}_{i=1}^{N}$, and for each sample, the CAD-based inertia matrix $M^{\text{CAD}}(q_i)$ is computed. Then, the joint distribution of $[q, M]^\top$ is modeled using a PLM, and the model parameters—including the cluster means $\{\mu_k^q\}_{k=1}^{C}$ (i.e., the centers of the local models)—are optimized so that the conditional regression $\hat{M}(q)$ approximates $M^{\text{CAD}}(q)$ well. In this study, the following cost is minimized:

$$e = \sum_{i=1}^{N} (M^{pred}(\boldsymbol{q}_i) - M^{CAD}(\boldsymbol{q}_i))^2. \tag{7}$$

Here, $M^{pred}(q_i)$ denotes the inertia matrix predicted by the PLM. To improve numerical stability and ensure the preservation of symmetric properties, the optimization is carried out using upper-triangular vectorization ($\text{vec}_u M$).

The number of clusters C is selected using a criterion commonly employed in GMM to avoid both overfitting and underfitting. Specifically, for each candidate C, the Akaike Information Criterion (AIC) is calculated, which considers both the fit to the above cost (log-likelihood) and the number of parameters [23]. The resulting μ^{opt} can be interpreted as the set of centers that most efficiently partitions the landscape of $M(q)$. It naturally captures physically significant regions across the entire workspace, such as areas with abrupt changes in inertial coupling or near workspace boundaries.

2) Stage 2: OPCET generation: Stage 2 aims to design a trajectory that ensures the state $q(t)$ visits the vicinity of each optimal local model position (μ_k^{opt}) obtained in Stage 1. The key lies in the construction of the cost function. First, the term l_m encourages each local model of the PLM, later trained using the EM method, to align with the μ_k^{opt} determined in Stage 1. The term l_r ensures that the physical constraints of the robot are satisfied, while l_s enhances the statistical quality by promoting a velocity–acceleration distribution that is voluminous, isotropic, and low in correlation.

To this end, the following cost function is minimized with respect to the trajectory, which is parameterized by a multi-sine based parameter vector ϕ:

$$\min_{\boldsymbol{\phi}} \quad l_m(\boldsymbol{\phi}) + w_r l_r(\boldsymbol{\phi}) + w_s l_s(\boldsymbol{\phi}). \tag{8}$$

The primary aim of the Stage 2 optimization is to ensure that the cluster centers selected in Stage 1 will coincide with the actual cluster means of the trajectory data generated in Stage 2. Therefore, the term l_m is designed to achieve this goal. The term l_m consists of three complementary components: l_μ, which aligns the mean of the local model with the optimal mean; l_{ℓ_2}, which ensures alignment with the global mean; and l_{nll}, which is composed of a negative log-likelihood to maximize the mixture likelihood.

First, the formula for l_μ is as follows.

$$l_\mu(\boldsymbol{\phi}) := \frac{1}{C} \sum_{k=1}^{C} \big\| \widehat{\boldsymbol{\mu}}_k(\boldsymbol{\phi}) - \boldsymbol{\mu}_k^{\text{opt}} \big\|_2^2,$$

$$\text{where} \quad \widehat{\boldsymbol{\mu}}_k(\boldsymbol{\phi}) = \frac{1}{N_k + \varepsilon} \sum_{i=1}^{N} h_k(q_i; \psi^{opt}) q(t; \phi), \tag{9}$$

$$N_k = \sum_{i=1}^{N} h_k(q_i; \psi^{opt}).$$

Using the $\psi^{opt} = [\pi_k, \mu_k^{opt}, \Sigma_k]_{k=1}^{C}$ obtained in Stage 1 for

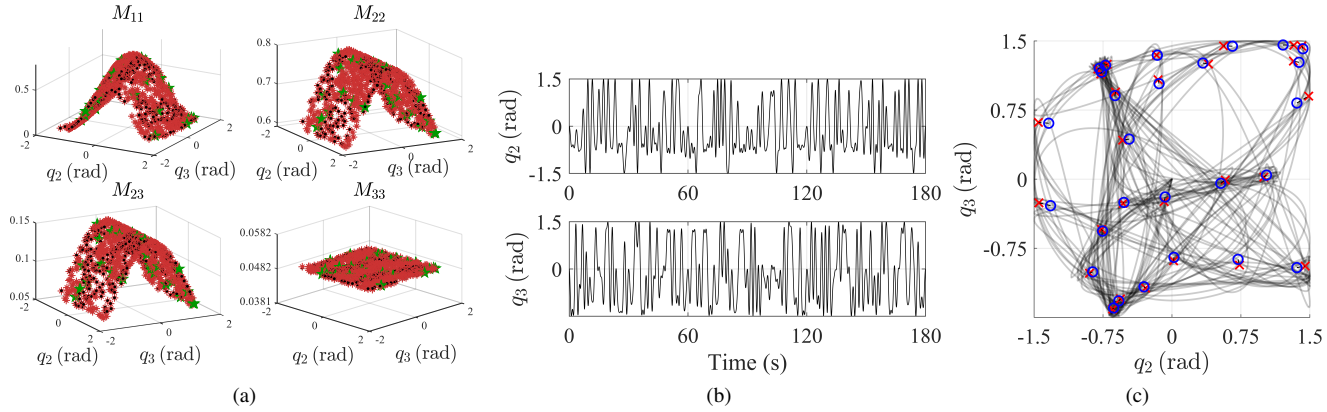

Fig. 3. (a) Result of Stage 1 (Algorithm 1) — (black: sample point, red: inertia component GMR results, green star: optimized mean points). Inertia matrix regression result: RMSE 5.64e-8, (b) Result of Stage 2 (Algorithm 2) — OPCET, (c) GMM result of the generated trajectory — (red x: optimal mean point (target), blue o: actual cluster means of the generated trajectory in (b), gray: trajectory). It can be seen that the cluster means are located close to the optimal mean points found in Stage 1 (RMSE: 0.0674 rad).

each local model as a fixed mixture model, the computed $h_k(q_i;\psi^{opt})$ is used as a weight. In this way, the weighted average $\hat{\mu}_k(\phi)$ of the trajectory samples assigned to component k is driven to converge to the target mean μ_k^{opt}. Compared to l_μ, the term l_{ℓ_2} provides a gentler pull, encouraging the overall dataset's mean to be close to the set of optimal means μ_k^{opt}. The formula is as follows.

$$l_{\ell_2}(\phi) = \left\|\overline{q}(\phi) - \mu^{opt}\right\|_2^2. \tag{10}$$

The term l_{nll} encourages the generated trajectory to more frequently fall into high-likelihood regions (peaks) under the fixed mixture distribution defined by ψ^{opt}. While l_μ determines the location of the local model, l_{nll} plays the role of guiding the distribution to match the area around that aligned location.

$$l_{nll}(\phi) = -\frac{1}{N}\sum_{i=1}^{N}\log\left(\sum_{k=1}^{C}\pi_k\mathcal{N}\left(q_i;\mu_k^{opt},\Sigma_k\right)\right). \tag{11}$$

Through l_m, when PLM modeling is performed on the generated trajectory, the local model can be positioned at the intended optimal location.

Secondly, l_r applies a soft penalty that gently pushes the endpoints toward the boundaries to ensure full coverage of the joint space's RoM. Rather than simply acting as a constraint on the range, it encourages the tails of the distribution to reach near the boundaries, so that the generated trajectory can cover the entire RoM.

Finally, l_s aims to ensure statistical quality in dynamics modeling using PLM by simultaneously achieving an expansion of the velocity-acceleration distribution volume (l_{vol}), improved isotropy (l_{iso}), and decorrelation between q and $\dot{q}$, as well as between q and $\ddot{q}$ (l_{decol}). This supports the rationale for focusing on $M(q)$ as discussed in Eq. (6) and in Stage 1. For the l_{vol} term, letting $Z_{va}(\phi) = [\dot{q}, \ddot{q}]$, the formula is constructed as follows.

$$l_{vol}(\boldsymbol{\phi}) = -\log\det\left(\mathrm{cov}\left(Z_{va}(\phi)\right) + \varepsilon I\right). \tag{12}$$

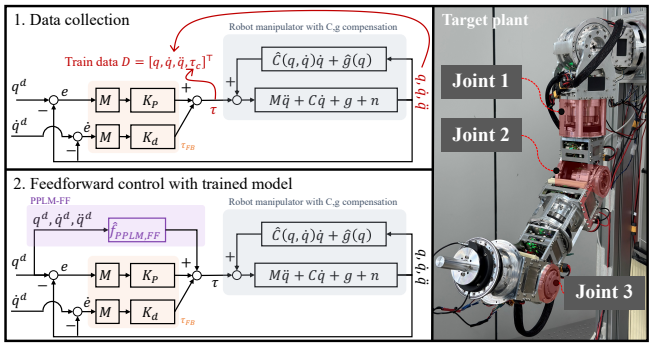

Fig. 4. Training data collection, control process and target plant. $\hat{f}_{PPLM,FF}$ is feedforward inverse inertia model.

l_{vol} increases the volume of the distribution by minimizing the negative log-determinant of the covariance matrix, allowing it to spread along the principal directions. l_{iso} enhances isotropy by balancing the variance of velocity, $\mathrm{var}(\dot{q}) = v_1$, and the variance of acceleration, $\mathrm{var}(\ddot{q}) = v_2$. This prevents the information from being biased toward specific cases (e.g., when only acceleration is large or only velocity is large). The specific formula is as follows.

$$l_{iso}(\phi) = \left(\frac{v_1 - v_2}{v_1 + v_2 + \epsilon}\right)^2, \tag{13}$$

l_{decol} reduces the correlations between q and $\dot{q}$, as well as between q and $\ddot{q}$, by minimizing $\rho_{q,\dot{q}}$ and $\rho_{q,\ddot{q}}$ simultaneously in the correlation matrix of the normalized $[q, \dot{q}, \ddot{q}]$. The specific formula is as follows.

$$\begin{aligned} &l_{\mathrm{decol}}(\boldsymbol{\phi}) = \rho_{q,\dot{q}}^2 + \rho_{q,\ddot{q}}^2, \\ &\rho_{q,\dot{q}}, \rho_{q,\ddot{q}} \quad \text{from } \mathrm{cov}\big(\text{z-score}([\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}])\big). \end{aligned} \tag{14}$$

In summary, the Stage 2 process directly guides the GMM local model means to align with the operating points from Stage 1 through l_m, ensures efficient use of the RoM via l_r, and guarantees the statistical quality of velocity–acceleration (large volume, isotropy, low correlation) through l_s. At the

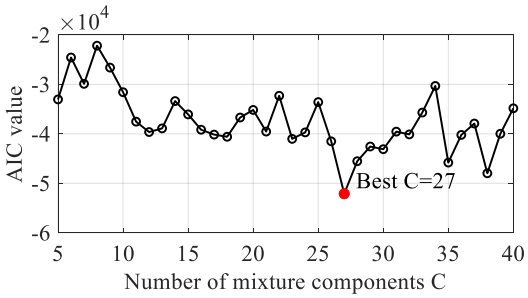

Fig. 5. A diagram illustrating the case where the optimal number of local models C is determined using AIC. For the 3-DoF manipulator, AIC values are iteratively recorded by applying the Stage 1 algorithm with a minimum of 5 and a maximum of 40 local models. The value $C = 27$, which corresponds to the lowest AIC, is selected as the optimal number of local models.

same time, velocity/acceleration limits and boundary conditions are enforced as hard constraints, making the generated trajectory a dynamically valid excitation signal that can be used immediately for actual robot operation.

IV. Validation of Proposed Training Data Generation Algorithm

A. Setup and Fair Comparison Protocol

The CATALYST framework was verified using a 3-DoF yaw–pitch–pitch manipulator. To this end, a 3-DoF model was constructed by simplifying the URDF of the existing 7-DoF ExSLeR rigid-actuator manipulator (Fig. 4) [24]. The reduction to 3 DoF was intended to improve computational efficiency and to enable clearer structural analysis during the data generation process. In particular, this simplified structure offers the advantage of allowing the inertia matrix to be visualized and intuitively analyzed, making it easier to evaluate the performance of the proposed algorithm and to understand its dynamic characteristics. The simulations were conducted in the MATLAB Simulink environment, using an Intel Core i5-12400F (12th Gen) CPU and an NVIDIA GeForce RTX 3060Ti GPU as computational resources. The physical model was implemented based on Simscape Multibody, and simulations were carried out using the variable-step ODE15s (stiff/NDF) solver. The controller was configured to operate at a 1 kHz fixed-step interval, reflecting an actual controller execution environment.

The validation in this study consists of four baselines: Spread/RoM covering, ill-centered, Tukey-windowed chirp, and Cubic. For a fair comparison, all baselines use the same data size, defined as $D = [q, \dot{q}, \ddot{q}, \tau] \in \mathbb{R}^{180001\times 12}$. A total of 180,001 data points is obtained by sampling for 180 seconds at 1 kHz, ensuring that sufficient physical information is captured.

1) Spread/RoM covering training trajectory: The Spread/RoM covering dataset is generated using a cost function with all l_m terms removed, so no bias toward optimal means exists. This encourages the trajectory to uniformly cover the RoM while still respecting physical constraints. This allows the learning target to explore a wide range of motion while still complying with physical constraints such as the RoM.

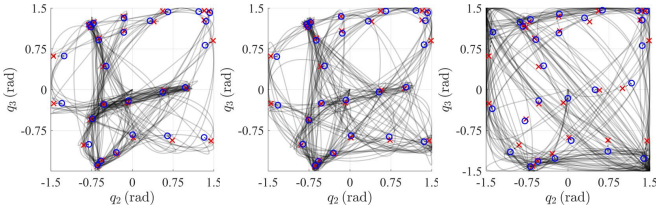

Fig. 6. Influence of l_s. (Left): $w_s = 0$, RMSE: 0.081, (Center; selected for paper) $w_s = 0.2$, RMSE: 0.067, (Right) $w_s = 2.0$, RMSE: 0.144. Marker information same as Fig. 3 (c)

2) ill-centered training trajectory: The ill-centered dataset simulates omission of Stage 1 (no guided mean selection). The trajectory still spans the whole RoM, but without targeted operating points the resulting cluster centers end up poorly placed. While the data itself covers the entire range, the absence of reference points (optimal mean points) results in poorly aligned local modeling.

3) Tukey-windowed chirp training trajectory: The Tukey-windowed chirp data is constructed by applying a Tukey window to a chirp signal, creating a waveform in which frequency components are gradually introduced and tapered [25].

4) Cubic training trajectory: Lastly, the Cubic dataset uses a cubic-spline reference (100 waypoints covering the RoM). This represents a case of task-centric data collection that intentionally ignores any physics-based considerations.

These baselines reflect widely used trajectory-generation heuristics in the context of GMM/GMR modeling; while Fisher/PE and active exploration (ASID, DAgger) are valuable in other settings, they target parametric identification or policy adaptation, and are therefore less directly comparable to our physics-guided nonparametric PLM approach.

B. Performance Verification of CATALYST on a 3-DoF Manipulator

1) Generated trajectory quality analysis for each stage: Fig. 3 shows the results of applying the proposed 2-stage training data generation algorithm, CATALYST, to a 3-DoF manipulator. In this example, it is assumed that the RoM of each joint is ± 1.5 rad.

Fig. 3 (a) uses the inertia matrix $M(q)$ obtained from the CAD model to generate sample data across the entire RoM. Considering the symmetry of the inertia matrix, only the upper triangular components are calculated (black dots). In the case of the 3-DoF yaw–pitch–pitch manipulator, only q_2 and q_3 are treated as variables, and q_1 is treated as independent; thus only q_2 and q_3 are analyzed [26].

We use the Akaike Information Criterion (AIC) to select the number of clusters. As shown in Fig. 5, the minimum AIC occurs at 27 components, which we adopt for the 3-DoF model.

Using the μ^{opt} obtained in Stage 1 as the optimal mean points for q_2 and q_3, reference trajectories are generated through the process of Algorithm 2. Since q_1 does not

TABLE I. Tracking Control Case 1: Cubic Reference

Type of training dataset	Torque error	Tracking error	FB torque
(Proposed) CATALYST	0.1039	0.0049	46.87%
Spread / RoM covering	3.1027	0.0768	48.59%
ill-centered	1.2748	0.0076	43.32%
Tukey-windowed chirp	0.2888	0.0061	45.67%
Cubic	0.1807	0.0031	49.06%

TABLE II. Tracking Control Case 2: Sine Reference

Type of training dataset	Torque error	Tracking error	FB torque
(Proposed) CATALYST	0.0431	0.1487	37.05%
Spread / RoM covering	0.0855	0.1500	41.41%
ill-centered	0.0747	0.1461	36.46%
Tukey-windowed chirp	0.1234	0.1483	57.23%
Cubic	0.0858	0.1441	56.03%

influence the inertia matrix (and thus has no target μ^{opt} from Stage 1), we generate its trajectory by optimizing only the l_r and l_s terms (omitting l_m). The resulting trajectory (Fig. 3 (b)) satisfies the RoM limits and all imposed constraints

Fig. 6 shows the distribution of the generated OPCET and the corresponding PLM modeling results as a function of the weight w_s of l_s in Stage 2. As w_s increases, the variance of $\dot{q}$ and $\ddot{q}$ becomes larger, resulting in a more spread-out distribution of the OPCET. This confirms that the design of l_s term in Algorithm 2 has been successfully implemented as intended.

After generating q_2 and q_3, GMM modeling is performed on $[q_2, q_3]^\top$, and it can be observed that the 27 clusters have means located at the blue-circle positions in Fig. 3 (c). The learned cluster centers (blue circles in Fig. 3 (c)) lie very near the Stage 1 target points (red X marks), with an RMSE of only 0.0674 rad.

Thus, the CATALYST framework shows that the algorithms in both Stage 1 and Stage 2 are well-designed to achieve their intended objectives.

2) PPLM training process with OPCET: Since PPLM aims to model Eq. (6), it models the robot's motion using measurements of $[q, \dot{q}, \ddot{q}, \tau]^\top$. This can be mathematically expressed as $M(q)\ddot{q} + N(q, \dot{q}, \ddot{q}) = \hat{f}_{PPLM}(q, \dot{q}, \ddot{q})$.

The trajectory generated through CATALYST framework is used as the reference trajectory $[q^d, \dot{q}^d, \ddot{q}^d]^\top$ for the robot under Computed Torque Control (CTC), with $C(q, \dot{q})$ and $g(q)$ compensated. During tracking, the measured $[q, \dot{q}, \ddot{q}]^\top$ and τ are used to model $\hat{f}_{PPLM}(q, \dot{q}, \ddot{q}) = \tau$. The learned $\hat{f}_{PPLM}$ can then be used in control frameworks that require a nominal model, as in the case shown in Fig. 4.

C. Comparison results

The CATALYST framework aims not only at dynamics identification but also at dynamics modeling of robot manipulators for use in model-based feedforward control and Disturbance Observers (DOB). Therefore, it is important to compare the control performance of PPLM from a control perspective. In this paper, as shown in Fig. 4, the designed PLM is used as the nominal model for feedforward control in joint tracking of a robot manipulator, and comparisons are made with the baselines.

First, a separate test reference trajectory is prepared. Case 1 uses a cubic trajectory as the reference, and testing is conducted with three different cubic trajectories, each having distinct waypoints. Each of these has a data size of $q^{d,cubic} \in \mathbb{R}^{10001\times 3}$. Case 2 uses sinusoidal signals based on either the 4^{th} or 5^{th} order Fourier components as the reference, and similarly, testing is conducted with three trajectories, each having different frequencies. The data size for each is the same as in case 1.

In Cases 1 and 2, the min-max ranges of the trajectories differ (Case 1: ± 0.5 rad, Case 2: ± 1 rad), as do the distributions of velocity and acceleration, resulting in different torque min-max ranges as well (Case 1: ± 1.5 Nm, Case 2: ± 20 Nm). Therefore, normalized root-mean-square error (nRMSE) is used as the evaluation metric for both torque regression and reference angle tracking.

$$\text{nRMSE} := \frac{1}{y_{max} - y_{min}} \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}. \tag{15}$$

A lower feedback control torque contribution indicates a higher contribution from the feedforward torque, which in turn implies that the learned model is closer to the actual system model. Additionally, as the role of feedback decreases and that of feedforward increases, control stability and control frequency are improved. We quantify this as:

$$C_{FB} := \frac{\sum_{i=1}^{N} \|\tau_{FB}(t_i)\|_2^2}{\sum_{i=1}^{N} \|\tau_{FB}(t_i)\|_2^2 + \sum_{i=1}^{N} \|\tau_{FF}(t_i)\|_2^2} \cdot 100[\%] \tag{16}$$

1) Reference tracking performance and Feedback control torque contribution comparison: Evaluating reference tracking performance is important, as tracking is one of the primary objectives of control. Since the control method involves both feedforward and feedback components, it is helpful to consider both tracking performance and feedback control torque contribution together for a clearer interpretation.

In Case 1, although the CATALYST-trained model had the lowest torque prediction error, the Cubic model achieved a slightly lower tracking error (position nRMSE 0.0031). However, since the torque contribution for the cubic method was 49.06%, this suggests that the improved tracking performance was mainly due to the feedback component, rather than accurate modeling.

Case 2 shows a similar pattern. The cubic method again achieved better tracking performance than the proposed framework, but with a high feedback torque contribution of 56.03%. This can be interpreted in the same way as Case 1.

Therefore, in the case of the CATALYST framework, the fact that it achieves both lower feedback torque contribution and strong tracking performance compared to other methods indicates that the dynamics modeling is closely aligned with the actual system.

2) Overall result analysis: In the case of the first baselines—Spread/RoM covering and ill-centered approaches—it becomes clear, considering the modeling method of PLM, that simply having data spread across the RoM does not necessarily result in accurate modeling of the entire space. Particularly in the ill-centered case, data tends to be concentrated in a specific region, causing the local model to also cluster in that area. As a result, both of these cases show inferior performance compared to the proposed framework. This highlights the importance of identifying physically significant regions throughout the entire domain and ensuring that the local model is positioned in those regions. In the case of the cubic method further highlights the risk of focusing solely on task kinematics when constructing training data for learning dynamics.

Ultimately, as seen in the CATALYST framework, analyzing both the physics-based priors of the system and the learning model comprehensively when collecting and training on data plays a crucial role in embedding physical information into the learning process.

V. Conclusion

This paper proposed the CATALYST framework, which incorporates physics-based priors directly into the design of training data—beyond only the model or loss function—for learning robot manipulator dynamics in a control-oriented manner. In a 3-DoF simulation, the proposed PPLM showed lower torque nRMSE and reduced feedback torque contributions compared to baselines (Spread/RoM, ill-centered, chirp, and cubic), while the learned cluster means closely aligned with the target operating points. These results suggest that physics-guided center placement and trajectory synthesis can compensate for a key limitation of EM-based PLM modeling—its sensitivity to data density—and enable the acquisition of control-friendly models under limited data budgets.

Future work will further strengthen robustness and practicality by validating on higher-DoF systems and real hardware, examining sensitivity to modeling mismatch and non-ideal effects, and developing a more systematic procedure for tuning objective weights. We also plan to deploy the CATALYST-trained PPLM as a nominal model in advanced controllers and disturbance observers.

References

[1] N. Jaquier, L. Rozo, D. G. Caldwell, and S. Calinon, "Geometry-aware manipulability learning, tracking, and transfer," *The International Journal of Robotics Research*, vol. 40, no. 2-3, pp. 624–650, 2021.

[2] M. Lutter and J. Peters, "Combining physics and deep learning to learn continuous-time dynamics models," *The International Journal of Robotics Research*, vol. 42, no. 3, pp. 83–107, 2023.

[3] M. Memmel, A. Wagenmaker, C. Zhu, P. Yin, D. Fox, and A. Gupta, "Asid: Active exploration for system identification in robotic manipulation," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024, oral.

[4] E. Daş and J. W. Burdick, "An active learning based robot kinematic calibration framework using gaussian processes," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 495–11 501.

[5] C. Frank, *Modern Robotics-Mechanics, Planning, and Control*. Cambridge University Press, 2017.

[6] J. Swevers, W. Verdonck, and J. De Schutter, "Dynamic model identification for industrial robots," *IEEE control systems magazine*, vol. 27, no. 5, pp. 58–71, 2007.

[7] E. Wernholt and S. Gunnarsson, "Estimation of nonlinear effects in frequency domain identification of industrial robots," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 4, pp. 856–863, 2008.

[8] C. Rucker and P. M. Wensing, "Smooth parameterization of rigid-body inertia," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2771–2778, 2022.

[9] A. Albu-Schaffer, W. Bertleff, B. Rebele, B. Schafer, K. Landzettel, and G. Hirzinger, "Rokviss-robotics component verification on iss current experimental results on parameter identification," in *Proceedings IEEE International Conference on Robotics and Automation*, 2006, pp. 3879–3885.

[10] E. Rueckert, M. Nakatenus, S. Tosatto, and J. Peters, "Learning inverse dynamics models in o (n) time with lstm networks," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 811–816.

[11] J. Schreiter, P. Englert, D. Nguyen-Tuong, and M. Toussaint, "Sparse gaussian process regression for compliant, real-time robot control," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2586–2591.

[12] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey," *Cognitive processing*, vol. 12, pp. 319–340, 2011.

[13] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 2, pp. 286–298, 2007.

[14] N. K. Sinha, "System identification—theory for the user," *Automatica*, vol. 25, no. 3, pp. 475–476, 1989.

[15] T. Lee, B. D. Lee, and F. C. Park, "Optimal excitation trajectories for mechanical systems identification," *Automatica*, vol. 131, p. 109773, 2021.

[16] 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.

[17] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.

[18] T. Cederborg, M. Li, A. Baranes, and P.-Y. Oudeyer, "Incremental local online gaussian mixture regression for imitation learning of multiple tasks," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 267–274.

[19] S. Calinon, D. Bruno, and D. G. Caldwell, "A task-parameterized probabilistic model with minimal intervention control," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 3339–3344.

[20] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[21] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[22] T. Kim, J. Jeong, T. Kong, H. Lee, and S. Oh, "A novel approach for efficient gaussian mixture model using dynamics-motivated optimal excitation," in *2024 IEEE 33rd International Symposium on Industrial Electronics (ISIE)*. IEEE, 2024, pp. 1–4.

[23] Y. Sakamoto and H. Akaike, "Analysis of cross classified data by aic," *Annals of the Institute of Statistical Mathematics*, vol. 30, no. 1, pp. 185–197, 1978.

[24] D. Lee, K. Choi, J. Kim, W. Yun, T. Kim, K. Nam, and S. Oh, "Exsler: Development of a robotic arm for human skill learning," in *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2023, pp. 209–214.

[25] P. Pallav, T. H. Gan, and D. A. Hutchins, "Elliptical-tukey chirp signal for high-resolution, air-coupled ultrasonic imaging," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 54, no. 8, pp. 1530–1540, 2007.

[26] K. Choi, J. Song, W. Yun, D. Lee, and S. Oh, "Identification of flexible joint robot inertia matrix using frequency response analysis," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 8704–8710.