Robot Learning from Multiple Demonstrations Based on Generalized Gaussian Mixture Model

Jiale Dong, Weiyong Si, Ning Wang, Chenguang Yang*

***Abstract*— The quality of skills learned by robots from demonstrations depends on the level of demonstration by the instructor, while models such as Dynamic Motion Primitives (DMP) are extremely sensitive to the quality of demonstrations because they can only learn from a single demonstration. In this paper, we propose a method of learning from multiple demonstration. First, we model multiple demonstration data by generalized Gaussian mixture models (GMMs), then generate a smooth trajectory by the Gaussian mixture regression (GMR), and finally, the generated trajectory is modeled by DMP. This method can capture features from multiple demonstrations of the same task, reducing the impact of incorrect demonstrations on skill learning. Additionally, the generalized Gaussian mixture model has strong data fitting ability, enabling robots to learn smooth and high-quality motion skills. Finally, the effectiveness of the proposed method was verified by performing a Chinese character writing task on a Baxter robot.**

I. Introduction

Robots are usually used to complete fixed tasks in structured environments, such as automobile manufacturing, electronic product assembly, etc. With the continuous and rapid development of the robotics industry, in the future, more and more robots will provide services for human daily life, such as nursing robots [1], [2] and cleaning robots. In a structured factory environment, task trajectories are usually carefully designed and coded for robots, where robots only need to repeat tasks according to predetermined trajectories. However, in unstructured environments, pre-designed trajectories cannot cope with dynamic and unknown environments, nor can they adapt to changes in tasks. In order to make it easier for robots to work in the actual environment, it is a simple and effective method to transfer human skills to robots. Learning from demonstration (LfD) [3] has received widespread attention in recent years [4]. It is an effective method to reduce the complexity of Robot learning new skills. Human instructors only need to demonstrate how to complete a new task, and robots can learn motor skills from the demonstration. There are three core issues in the learning of robot motor skills. One is the stability of motion, which means ensuring that the motion converges to the target position. The second is the generalization ability of skills. We hope that the learned skills can be applied to different environments and tasks, rather than simply repeating the demonstration trajectory of the instructor. The third is the quality of motor skills. We hope that the movements learned by Robot learning are smooth and continuous.

At present, there are many methods used for motion trajectory modeling, such as spline decomposition [5], Gaussian mixture models (GMMs) [6], [7], hidden Markov model (HMM) [8], dynamic motion primitives (DMP), and so on. Using spline curves to encode trajectories can maintain their smoothness and continuity, and can flexibly represent and process various trajectory shapes. Both GMMs and HMM are statistical models used to describe the probability distribution of trajectory points, but they focus on different aspects. GMMs are more suitable for trajectory pattern recognition and clustering, while HMM is more suitable for temporal modeling and trajectory sequence generation. DMP motion as a dynamic system with an attractor. Its basic idea is to generate and control motion by decomposing complex motion into simple basic modes and using appropriate parametric representation. At present, DMP has been widely applied in the field of robotics [9], such as robot table tennis [10], robot online obstacle avoidance [11], and dual arm collaborative tasks [12]. Some researchers have also modified the original model [13], [14] to make it more scalable and generalized. [15]proposed a method that combines DMP with probability models, which can generate smooth trajectories and capture higher-order statistical information of motion. In addition to modeling motion trajectories, DMP can also model information such as force and stiffness obtained from demonstrations [16].

In the process of human learning, humans can master motor skills through multiple exercises. for learning from demonstration, the quality of skills learned by Robot often depends on the quality of demonstration data. We hope that robots can learn high-quality motion skills from some imperfect demonstrations, just like humans. Many researchers have studied the trajectory modeling method based on GMMs [17]. By GMMs, multiple demonstration data modeling can be modeled as a mixture of a group of Gaussian component. Each Gaussian component represents the behavior mode and state of the task trajectory. GMMs can extract the invisible features in the motion trajectory. [18] replace the nonlinear term in DMP with GMMs, so that they can learn from multiple demonstrations. However, trajectory models based on GMMs have low computational efficiency and poor modeling performance for nonlinear data. In this article, we propose a method of learning skills from multiple demonstrations based on Generalized Gaussian Mixture model. Firstly, a generalized Gaussian mixture model [19], [20] is used to model the demonstration data. Compared to conventional GMMs, the

J. Dong is with the School of Automation Science and Engineering, South China University of Technology, Guangzhou, China.

W. Si, N. Wang and C. Yang are Bristol Robotics Lab at the University of the West of England, Bristol, BS16 1QY, UK.

* Corresponding author is C. Yang (Email: cyang@ieee.org)

generalized Gaussian mixture model has better modeling performance for nonlinear data. Then, we use the corresponding Gaussian mixture regression algorithm [21] to obtain a smooth trajectory. Finally, model the generated trajectory by DMP. In this way, we can learn a relatively perfect trajectory from multiple demonstrations, and use DMP to model the trajectory, making motion skills have strong generalization performance and can achieve real-time motion generation.

The main contributions of this paper are twofold. Firstly, we propose a learning method based on the generalized Gaussian mixture model, Gaussian mixture regression, and DMP. Secondly, we adjust the original Gaussian mixture regression algorithm to make it applicable to the generalized Gaussian mixture model, too. The rest of this article is organized as follows. The second section introduces DMP, generalized Gaussian mixture model, and Gaussian mixture regression. The third section presents simulation and experimental results, and the fourth section presents conclusions.

II. Methodology

A. Dynamic Movement Primitive

Dynamic Motion Primitives (DMP) is essentially a second-order nonlinear equations that can be regarded as spring damping systems driven by nonlinear terms. DMP consists of a linear term and a nonlinear term: the linear term is a function of time, which is related to the manually set target trajectory. Its function is to make the robot follow the set target trajectory when executing motion. The linear term usually takes the form of a time proportional factor, which gradually decreases over time. The form of the nonlinear term allows the robot to adjust the speed and amplitude of motion adaptively, so that the motion track shows a specific shape. The original DMP can be represented by the following equation [22]:

$$\tau\dot{v} = k(g - x) - dv + f(s) \tag{1}$$

$$\tau\dot{x} = v \tag{2}$$

$$\tau\dot{s} = -\alpha_1 s \tag{3}$$

Where x and v represent the position and velocity of the demonstration trajectory, respectively, g is the target position, k is the stiffness coefficient of the system, d is the damping coefficient of the system, and s is the phase of the system, gradually decaying from the initial value of 1 to 0. DMP does not directly depend on time, but rather on phase, which makes it easy to expand the model, τ represents the time constant of the system, which determines the evolution time of the system and can be adjusted to change the duration of the task. α_1 is a normal number, and $f(s)$ is a nonlinear continuous bounded function driven by the system phase. When s decays to 0, $f(s)$ becomes a constant. The nonlinear term $f(s)$ is defined as:

$$f(s) = \frac{\sum_{i=1}^{N} \psi_i \cdot \omega_i}{\sum_{i=1}^{N} \psi_i} \cdot (g - x_0)s \tag{4}$$

$$\psi_i = exp(-h_i(s - c_i)^2) \tag{5}$$

Where x_0 represents the initial position of the trajectory, psi_i is the radial basis function, c_i and h_i represent the center and width of the radial basis function respectively, ω_i is the weight of the i_{th} basis function, and N is the number of basis functions, determined according to the task. The estimation of ω_i is a supervised learning process. After obtaining the demonstration trajectory, record its position $x(t)$ and at each time step t=0, 1, ..., p and calculate its acceleration and velocity $\ddot{x}(t)$, $\dot{x}(t)$, and obtain the objective function based on these data and Eq. (1):

$$f_{target}(s) = \tau\dot{v} + dv - k(g - x) \tag{6}$$

$$J = \sum (f_{target}(s) - f(s))^2 \tag{7}$$

Then, by minimizing the function J, the weights ω_i of the base function ψ_i is obtained. Usually, locally weighted regression (LWR) and locally weighted projection regression (LWPR) are used to solve this problem. The method of using locally weighted regression is as follows:

$$\omega_i = \frac{z^T \Gamma_i f_{target}}{z^T \Gamma_i z} \tag{8}$$

where,

$$\Gamma_i = \begin{pmatrix} \psi_i(t_1) & 0 & ... & 0 \\ 0 & \psi_i(t_1) & ... & 0 \\ 0 & 0 & ... & 0 \\ 0 & 0 & ... & \psi_i(t_P) \end{pmatrix} \tag{9}$$

The DMP has the following characteristics:

1) Strong robustness, able to resist uncertainty and interference during motion. From Eq. (3), it can be seen that the phase s will decay to 0 over time. Therefore, the nonlinear term $f(s)$ is bounded. It is only necessary to ensure that the system stiffness coefficient k and damping coefficient d are both greater than 0 to ensure that the system is stable and will eventually converge to the target position.
2) Has strong adaptability and generalization, by modifying the starting position x_0 of DMP and target position g can achieve generalization of trajectories in space, modify the time constant in the model τ, the generalization of trajectories in time can be achieved, and the shape of the generalized trajectory is roughly similar to the original trajectory.

B. Generalized GMMs

Classical GMMs only include conventional Gaussian models, as their axes are linear, so it usually requires more components when fitting nonlinear datasets. Active curve axis Gaussian model (AcaG) [20] has a bent principal axis, so its fitting effect on nonlinear data is better, as shown in Fig. 1. Next, we introduce generalized GMMs [19] that include both conventional Gaussian models and AcaG. Assuming $X = \{x_1, ..., x_n\}$ is a d dimensional observation dataset of n vectors, and the distribution of X is based on Gaussian

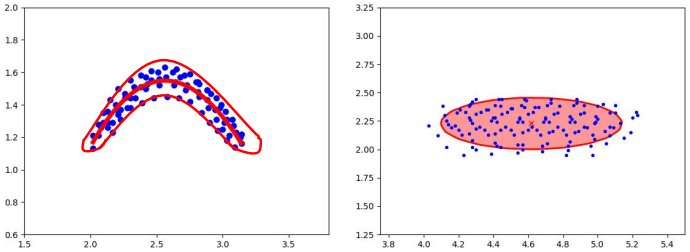

Fig. 1: The left is AcaG, and the right is conventional Gaussian model

component or AcaG, a generalized GMMs can be represented by the following equation:

$$p(x_t|\Theta) = \sum_{i=1}^{k} \alpha_i p_i(x_t|\theta_i) \tag{10}$$

where k is the number of Gaussian components, each component may be a conventional Gaussian model or AcaG, α_i is the weight of each Gaussian component, θ_i is the parameter of each Gaussian component, the process of calculating the parameters of each component in GMMs is as follows:

(1) Divide the data into K classes by k-means clustering, where K is pre-set, meaning there are a total of K Gaussian components, this is obtained by observation of the data, and then for each class of data, transforming raw data by PCA:

$$y_i = Q \times (x_i - T) \tag{11}$$

Where T and Q are the translation matrix and rotation matrix of PCA respectively, $Y = \{y_1, ..., y_m\}$ is the transformed coordinates of raw data, the d dimensions of Y are denoted as $v = [v_1, ..., v_d]^T$, in this paper, the data is all two-dimensional. Then, fit $[v_1, v_2]$ by the least squares method, and the standard curve which is defined as $y = ax^2 + b$, then we divide each component into a conventional Gaussian model and AcaG based on the degree of curvature of the main axis, a small positive real number ε is preset. If $|a| < \varepsilon$, it can be considered as the main axis approaching a straight line, at this point, the Gaussian component is a conventional Gaussian component. If $|a| \geq \varepsilon$, then it is considered the main axis is sufficiently curved, and the Gaussian component at this point is AcaG.

(2) Initialize the parameters of each component. For a conventional Gaussian model, the parameters are the center point and covariance matrix. For AcaG, the probability density function of point x_t is:

$$p(x_t|\theta) = \sum_{j=1}^{j_t} \prod_{s=1}^{2} \frac{exp(-l_j^2(v_{st})/2\Sigma_s)}{\sqrt{2\pi|\Sigma_s|}} \tag{12}$$

Where j_t is the number of projection points of x_t on the main axis, Σ_1 and Σ_2 represent the two covariances on the main axis, $l_j(v_{1t})$ is the arc length from the j_{th} projection point to the midpoint of the main axis, and $l_j(v_{2t})$ is the distance from x_t to the j_{th} projection point. The line connecting point x_t and the projection point is perpendicular to the tangent of the main axis at the projection point, as shown in Fig.2. The parameters of AcaG is $\theta = (\mu, \Sigma_1, \Sigma_2, C, Q, T)$, where μ is the center point, and C represents the parameters a and b of the principal axis curve. μ is initialized by k-means, Q and T are initialized by PCA, C is initialized by the least squares method, Σ_1 and Σ_2 are initialized by following equation:

$$\Sigma_1 = \frac{1}{n} \sum_{i=1}^{n} (l_{nearest}(v_{1i}))^2 \tag{13}$$

$$\Sigma_2 = \frac{1}{n} \sum_{i=1}^{n} (l_{nearest}(v_{2i}))^2 \tag{14}$$

where $l_{nearest}$ represents the projection point closest to x_t.

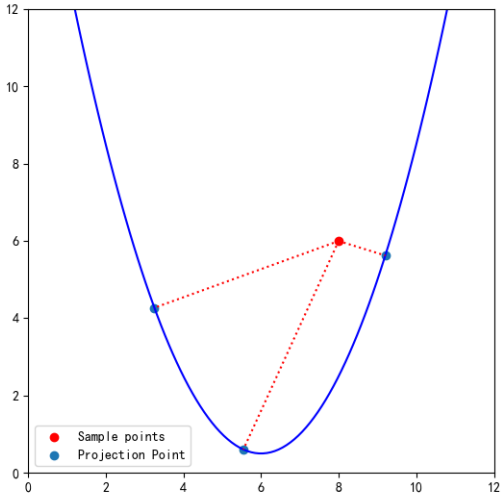

Fig. 2: Projection of sample points on the main axis

(3) Iteratively update the parameters of each component, the iterative method for the EM algorithm for generalized GMMs is as follows:

E step: Compute 'expected' component of all data points for each component.

$$w_{it} = \frac{\alpha_i p_i(x_t|\theta_i)}{\sum_{s=1}^{k} \alpha_s p_s(x_t|\theta_s)} \tag{15}$$

Where w_{it} is the posterior probability of x_t belonging to the i_{th} Gaussian component in generalized GMMs.

M step: Calculate the maximum likelihood of each Gaussian component for a given data. Then update the weight for each component:

$$\alpha_i^{new} = \frac{1}{n} \sum_{t=1}^{n} w_{it} \tag{16}$$

For AcaG, the update process of its parameters is as follows:

$$X_i^{'} = [w_{i1}, ..., w_{in}] \cdot [x_1, ..., x_n] \tag{17}$$

$$(C_i^{new}, T_i^{new}, Q_i^{new}) = LSFM(PCA(X_i^{'})) \tag{18}$$

$$\mu_i^{new} = \frac{\sum_{t=1}^{n} w_{it} x_t}{\sum_{t=1}^{n} w_{it}} + (Q_i^{new})^{-1}[0, b]^T + T_i^{new} \quad (19)$$

$$\bar{L}_{te}^{(i)} = \frac{\sum_{j=1}^{J_t^{(i)}} p(l_j^{(i)}(v_{et}) | N(0, \Sigma_{ie}^{old})) l_j^{(i)}(v_{et})}{\sum_{j=1}^{J_t^{(i)}} p(l_j^{(i)}(v_{et}) | N(0, \Sigma_{ie}^{old}))} \quad (e = 1, 2) \quad (20)$$

$$\Sigma_{ie}^{new} = \frac{\sum_{t=1}^{n} w_{it} \bar{L}_{te}^{(i)}}{\sum_{t=1}^{n} w_{it}} \quad (e = 1, 2) \quad (21)$$

where $PCA()$ is the principal component analysis function, and $LSFM()$ is the least squares fitting method. For more detailed descriptions of generalized GMMs and AcaG, as well as the process of parameter updates, please read [19] and [20].

C. Learning from Multiple Demonstrations

For the given demonstration data $D = \{p_{i,j}, t_{i,j}\}_{i=0,j=0}^{N,T_i}$, where p is the position, t is the time, N is the number of demonstrations, and T_i is the number of i_{th} demonstration sample points. First, model the demonstration data by the generalized GMMs that containing Kcomponents, Where K is set based on the trajectory demonstrated. Then, the expected maximum position ξ_p at a given time ξ_t can be calculated by GMR. The mean value of k_{th} Gaussian components in GMMs is:

$$\mu_k = \{\mu_{t,k}, \mu_{s,k}\} \quad (22)$$

The covariance matrix is:

$$\Sigma_k = \begin{pmatrix} \Sigma_{t,k} & \Sigma_{ts,k} \\ \Sigma_{st,k} & \Sigma_{s,k} \end{pmatrix} \quad (23)$$

For conventional Gaussian component, $\Sigma_{t,k}$ and $\Sigma_{s,k}$ are the variances of time and position, respectively. For the AcaG, $\Sigma_{t,k}$ and $\Sigma_{s,k}$ are the two covariances on the principal axis, for each Gaussian component, given ξ_t, the conditional expectation and estimated conditional covariance of ξ_s are:

$$\hat{\xi}_{s,k} = \mu_{s,k} + \Sigma_{st,k}(\Sigma_{t,k})^{-1} l(\xi_t, \mu_{t,k}) \quad (24)$$

$$\hat{\Sigma}_{s,k} = \Sigma_{s,k} - \Sigma_{st,k}(\Sigma_{t,k})^{-1}\Sigma_{ts,k} \quad (25)$$

For conventional Gaussian component, $l(\xi_t, \mu_{t,k}) = (\xi_t - \mu_{t,k})$, for AcaG, $l(\xi_t, \mu_{t,k})$ is the arc length from the intersection of $t = \xi_t$ and the main axis to the center point. Combine $\hat{\xi}_{s,k}$ and $\hat{\Sigma}_{s,k}$ based on the weight coefficients of each Gaussian component, for ξ_t:

$$\beta_k = \frac{p(\xi_t | k)}{\sum_{i=1}^{K} p(\xi_t | i)} \quad (26)$$

For GMMs with K Gaussian components, given ξ_t, the conditional expectation and conditional covariance of ξ_s are:

$$\hat{\xi}_s = \sum_{k=1}^{K} \beta_k \hat{\xi}_{s,k} \quad (27)$$

$$\hat{\Sigma}_s = \sum_{k=1}^{K} \beta_k^2 \hat{\Sigma}_{s,k} \quad (28)$$

Then, the corresponding trajectory can be obtained through a set of time series, and the generated trajectory can be modeled using DMP. Fig. 3 shows the process of learning from multiple demonstration groups. First, model multiple demonstration groups of data by generalized GMMs, then obtain a series of trajectory points by GMR regression to , and finally model the trajectory by DMP.

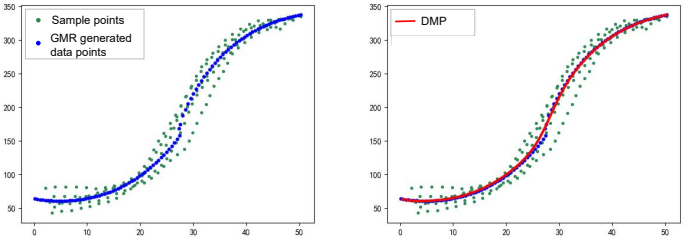

Fig. 3: DMP model for trajectory points and trajectory points generated by GMR.

D. Comparison of Learning from Multiple Demonstrations Based on Generalized GMMs and Conventional GMMs

We model multiple demonstrations by conventional GMMs and generalized GMMs respectively, and compare their fitting effects on nonlinear data. The effect of conventional GMMs on data modeling is shown in Fig. 4. The number of Gaussian components set in the three figures is 3, 5 and 10 respectively. It can be seen that if the number of models set is small, the fitting effect on data is poor. If there are too many models set, it is easy to cause overfitting. The effect of learning multiple

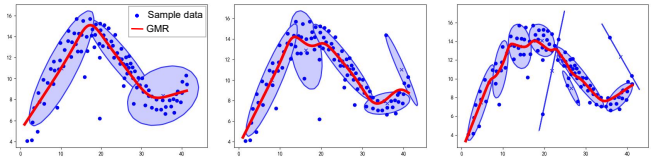

Fig. 4: Based on conventional GMMs-GMR learns multiple teaching data.

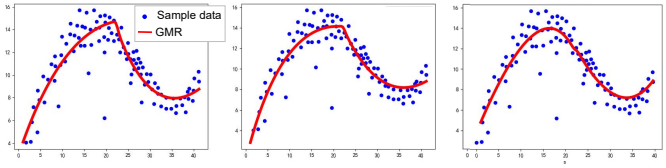

Fig. 5: Learning from multiple demonstration data based on generalized GMMs.

demonstrations by generalized GMMs is shown in Fig. 5. The set number of component fractions in the three images is 2, but the number of EM optimizations is different, namely 1, 3, and 10. As the number of iterations for optimization increases, the trajectory generated by GMR becomes smoother and smoother. It can be seen that compared to conventional GMMs, generalized GMMs performs better in fitting nonlinear data.

III. EXPERIMENT

We conducted writing experiments on Baxter robots to verify the effectiveness of the proposed method. As shown in Fig. 6, it is a schematic diagram of a robot writing Chinese characters.

Fig. 6: Robot writing Chinese characters.

A. Learning from demonstration

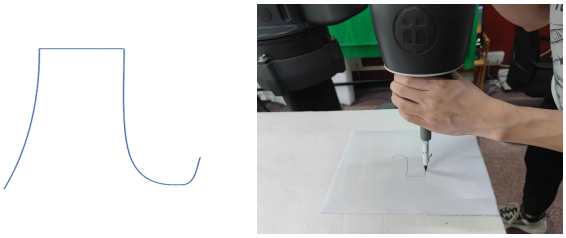

Fig. 7: Dragging the robotic arm to write Chinese characters.

As shown in Fig. 7, the instructor drags the robotic arm to complete the writing of Chinese characters on paper and records the trajectory of the end-effector during this process. During this process, dragging the robotic arm is a significant burden for the instructor, making it difficult to ensure the accuracy and smoothness of the writing trajectory. A total of three writing demonstrations were conducted, and the results and recorded data are shown in Fig. 8, it can be seen that the Chinese characters written by dragging are not smooth.

B. Skill learning

In the skill learning stage, we first use generalized GMMs to model the recorded data modeling. Since writing is carried out on a plane, we only focus on the tracks in the X and Y directions, and because the X and Y directions of Chinese characters are not one-to-one corresponding, it is not suitable for regression analysis, so we do not directly model the position tracks in both directions, but separately model the position and time in both directions. As shown in Fig. 9,

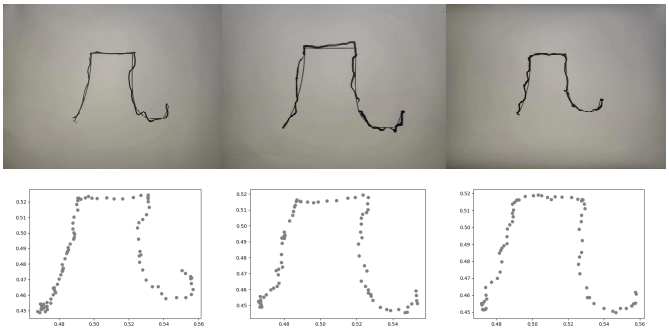

Fig. 8: The results of three writing sessions and the recorded end trajectory.

it is the position trajectory and time data in the X and Y directions. Then, two sets of data were modeled by generalized

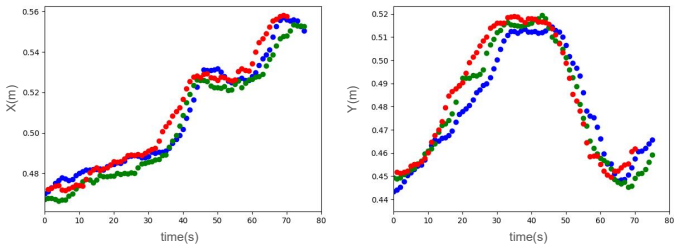

Fig. 9: The results of three writing sessions and the recorded end trajectory.

Gaussian mixture model. We used four Gaussian components for both sets of data, which were artificially set by observing the two sets of data, and the number of iterations for model optimization is 10. The results are shown in Fig. 10, which shows the principal axis of each component.

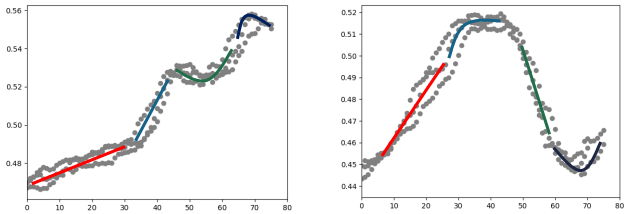

Fig. 10: The generalized GMMs model of the data, where the curve in the graph represents the principal axis of each Gaussian component.

After the model is established, given a series of times $T = t_1, t_2, ..., t_n$, we obtain the position trajectory corresponding to each time using GMR to generate a smooth trajectory. Then, we use DMP to generate the generated trajectory, and the results are shown in Fig. 11, it can be seen that the generated trajectory is very smooth.

C. Robot reproduction

After establishing the DMP in the X and Y directions, the desired trajectory for the robot to write Chinese characters

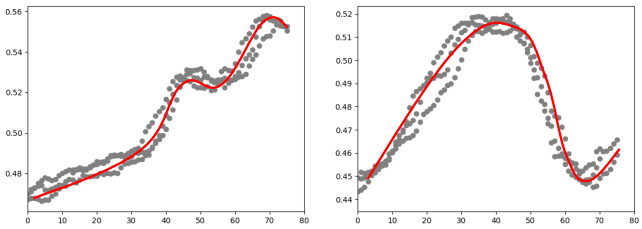

Fig. 11: DMP model based on GMR.

can be obtained, and then the robot can independently write Chinese characters based on the desired trajectory. Fig. 12 shows the expected trajectory for writing and the actual writing results of the robot, it can be seen that the Chinese characters written by robots are smoother than those manually dragged.

Although the demonstration data is not very smooth, generalized GMMs can extract all the features in the demonstration and present them in a smoother way, which fully demonstrates the necessity of learning skills from multiple demonstrations.

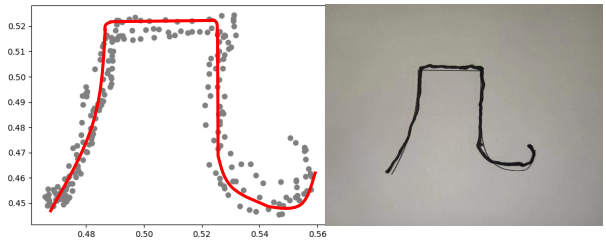

Fig. 12: The expected trajectory of writing and the results of autonomous robot writing.

IV. Conclusion

This paper proposes a method for learning skills from multiple demonstrations based on generalized GMMs. Generalized GMMs contain Active curve axis Gaussian, which enhances their fitting effect on nonlinear data. We compared the fitting effect of conventional GMMs and generalized GMMs on nonlinear data, demonstrating their superiority. Chinese character writing experiments were conducted on Baxter robots, and the optimized trajectory was smoother than the demonstration trajectory. However, there are still areas for improvement in the method presented in this article. As we model the position and time in each direction of the demonstration trajectory, it is necessary to match each trajectory in time. However, during multiple demonstrations, this is difficult to ensure. And the training time of generalized GMMs is long and the efficiency is very low. In future work, we will study how to solve these two problems.

References

[1] M. Laakasuo, J. Palomäki, A. Kunnari, S. Rauhala, M. Drosinou, J. Halonen, N. Lehtonen, M. Koverola, M. Repo, J. Sundvall *et al.*, "Moral psychology of nursing robots: Exploring the role of robots in dilemmas of patient autonomy," *European Journal of Social Psychology*, vol. 53, no. 1, pp. 108–128, 2023.

[2] D. Zhang, W. Si, W. Fan, Y. Guan, and C. Yang, "From teleoperation to autonomous robot-assisted microsurgery: A survey," *Machine Intelligence Research*, vol. 19, no. 4, pp. 288–306, 2022.

[3] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annual review of control, robotics, and autonomous systems*, vol. 3, pp. 297–330, 2020.

[4] W. Si, N. Wang, Q. Li, and C. Yang, "A framework for composite layup skill learning and generalizing through teleoperation," *Frontiers in Neurorobotics*, vol. 16, p. 840240, 2022.

[5] J. Aleotti and S. Caselli, "Robust trajectory learning and approximation for robot programming by demonstration," *Robotics and Autonomous Systems*, vol. 54, no. 5, pp. 409–413, 2006.

[6] E. Gribovskaya, S. M. Khansari-Zadeh, and A. Billard, "Learning nonlinear multivariate dynamics of motion in robotic manipulators," *The International Journal of Robotics Research*, vol. 30, no. 1, pp. 80–117, 2011.

[7] C. Chen, C. Yang, C. Zeng, N. Wang, and Z. Li, "Robot learning from multiple demonstrations with dynamic movement primitive," in *2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 2017, pp. 523–528.

[8] G. E. Hovland, P. Sikka, and B. J. McCarragher, "Skill acquisition from human demonstration using a hidden markov model," in *Proceedings of IEEE international conference on robotics and automation*, vol. 3. Ieee, 1996, pp. 2706–2711.

[9] W. Si, N. Wang, and C. Yang, "Composite dynamic movement primitives based on neural networks for human–robot skill transfer," *Neural Computing and Applications*, pp. 1–11, 2021.

[10] K. Mülling, J. Kober, O. Kroemer, and J. Peters, "Learning to select and generalize striking movements in robot table tennis," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 263–279, 2013.

[11] S. Grafakos, F. Dimeas, and N. Aspragathos, "Variable admittance control in phri using emg-based arm muscles co-activation," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2016, pp. 001 900–001 905.

[12] A. Gams, B. Nemec, L. Zlajpah, M. Wächter, A. Ijspeert, T. Asfour, and A. Ude, "Modulation of motor primitives using force feedback: Interaction with the environment and bimanual tasks," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 5629–5635.

[13] W. Si, N. Wang, and C. Yang, "Reactive and constrained motion primitive merging and adaptation," in *2021 26th International Conference on Automation and Computing (ICAC)*. IEEE, 2021, pp. 1–6.

[14] Z. Lu, N. Wang, Q. Li, and C. Yang, "A trajectory and force dual-incremental robot skill learning and generalization framework using improved dynamical movement primitives and adaptive neural network control," *Neurocomputing*, vol. 521, pp. 146–159, 2023.

[15] G. Li, Z. Jin, M. Volpp, F. Otto, R. Lioutikov, and G. Neumann, "Prodmp: A unified perspective on dynamic and probabilistic movement primitives," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2325–2332, 2023.

[16] C. Zeng, C. Yang, J. Zhong, and J. Zhang, "Encoding multiple sensor data for robotic learning skills from multimodal demonstration," *IEEE Access*, vol. 7, pp. 145 604–145 613, 2019.

[17] S. Calinon, T. Alizadeh, and D. G. Caldwell, "On improving the extrapolation capability of task-parameterized movement models," in *2013 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2013, pp. 610–616.

[18] C. Yang, C. Chen, W. He, R. Cui, and Z. Li, "Robot learning system based on adaptive neural control and dynamic movement primitives," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 3, pp. 777–787, 2018.

[19] Z. Ju and H. Liu, "Fuzzy gaussian mixture models," *Pattern Recognition*, vol. 45, no. 3, pp. 1146–1158, 2012.

[20] B. Zhang, C. Zhang, and X. Yi, "Active curve axis gaussian mixture models," *Pattern recognition*, vol. 38, no. 12, pp. 2351–2362, 2005.

[21] X. Yin and Q. Chen, "Learning nonlinear dynamical system for movement primitives," in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2014, pp. 3761–3766.

[22] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Trajectory formation for imitation with nonlinear dynamical systems," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, vol. 2. IEEE, 2001, pp. 752–757.