Title

Wenjin Xu, Chenguang Yang *Fellow, IEEE,*

Abstract—In our daily life, human can accomplish complex tasks with great ease by inspiring and generalizing a set of fundamental skills from the past. Notably, it would be outstanding for the robotic control systems to benefit from such brilliant learning capability. In the last decades, many researchers have paid their massive efforts to investigate the robotic motor control learning problems to provide reasonable decision-making and advanced control methods, such as sliding mode control, adaptive NN control and robust control, etc. In general, it is well known that the robot dynamic is hard to be modeled appropriately with the increasing complexity of manipulator's structure. Due to the excellent approximation capability of NN and fuzzy system, they have been extensively utilized for adaptive control systems to estimate the unknown dynamic of robot at any specific accuracy. Particularly, a notable method which incorporates the fuzzy logic to adaptive NN control has been widely adopted to address the problems of lacking automatic learning capabilities in fuzzy system. By considering the uniform boundedness and state constraints, He *et al.* proposed an adaptive fuzzy neural network (FNN) control to address the interaction problem of robot. In, a developed FNN control structure is employed to approximate the uncertain dynamic model of a dual-arm robot. Despite much progress achieved in model approximation, several other methodologies have been proposed to ensure the control performance, such as input saturation which can tackle the control problem related to input constraint, finite-time convergence control.

Index Terms—robotic

I. INTRODUCTION

DURING the past few years, increasing robotic manipulator techniques have been employed into industrial applications and our daily life. One of the most difficult problems in robotic research is how to achieve the human-like working manner in its workspace, such as distinctive capability of skills learning and friendly desirable interaction control. Such abilities are of great importance to enable robots to respond more intelligently and be able to fulfil more complex, dexterous and versatile tasks in various fields such as industrial applications and service areas.

Nevertheless, a key problem in the aforementioned control schemes which is tough to address is that the neural nodes for NN or the logic rules for fuzzy system are specified beforehand in terms of the expert experiences. Once the desired motion trajectory or working environment is changed, the original controller, unsurprisingly, may result in poor control tracking performance even irreversible consequence. In such case, the FNN controller, which is lack of generation ability from neural biological attribute point of view, needs to be redesigned to deal with the new tasks. Recently, a novel framework named broad learning system, which is developed from random vector functional-link neural networks (RVFLNN), has been proposed and extensively utilized for pattern recognition and classification. By taking the advantages of RVFLNN, BLS can offer acceptable generalization and efficient expansion performance through increasing the neural nodes dynamically. Herein, this inspires us to bring a forward solution which incorporates BLS to improve the generalizing ability of FNN.

As mentioned above, it is innate for humans to adjust the impedance and force of their arms skillfully while interacting with the unknown environments. From a biological point of view, humans can adapt their endpoint impedance of limbs through the central nervous system to achieve compliant interaction. Therefore, it would be crucial and significant to control robot in such way for a variety of social applications such as health care, assembly line and human-robot collaboration. In the literature, significant progress has been made to tackle such related issues. Generally, it is of great importance to establish a desired impedance model for impedance control parameters of which are difficult to choose from in practical. One of the most important parts of impedance control is to determine the desired impedance parameters. Nevertheless, most existing impedance control methods used fixed and manual designed impedance model. In order to improve the performance of compliant interaction under unknown environment, optimization of the impedance model should be taken into consideration. In, a desirable force regulation and tracking performance have been achieved by means of linear quadratic regulator (LQR), which is operated by considering a minimal cost function, to optimize the impedance parameters under a vital precondition of completely known environment dynamics. However, such algorithms seem too conservative to achieve a desired robot-environment interaction since they lack the ability of incorporating the unstructured and dynamic environments. Progress has been made in addressing such problems, and various optimal control schemes in the case of unknown environment have been proposed in literature. It proposed a force sensorless admittance control for robots to tackle with the unknown robot-environment interaction.

The main contributions of this article are listed as follows. [1].

1) We propose a learning method based on DMP and FGMM which can not only learn the trajectory from multiple demonstrations but also learn the deviation among them.

2) We present a new LfD method which can demonstrate by a single static image, each sample point on the image is encoded by a prior trajectory and FGMM. The skills are then learnt through the proposed methods.

3) We give a method for generating a priori trajectories of Chinese characters and design a robot learning framework for learning to write. Experiments were carried out on the LASA dataset and Chinese character images to verify the effectiveness of the method.
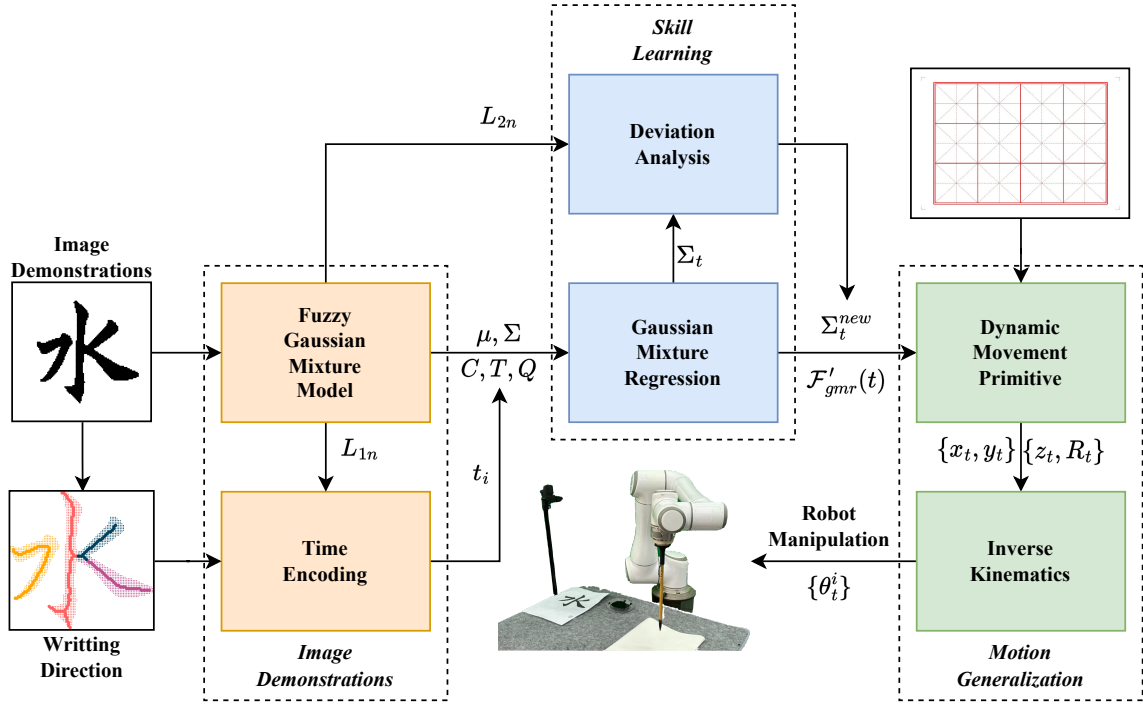
Fig. 1. Block diagram of the proposed system.

II. METHODOLOGY

A. System Description

The block diagram presented in Fig. 1 illustrates the framework of the imitation learning system for robotic handwriting tasks. The system can learn skills from static images and prior trajectories, particularly focusing on the deviation information within them, and ultimately achieving skill generalization in various environments. The entire system encompasses three aspects:

1) Image Demonstration: The demonstration part transfers the writing skill to the robot through the character image written by a human. The proposed system enables robot learning parameters of FGMM from a static image. On the other hand, the priori information of time is attained by the priori trajectories, which explicitly show the implicit time information in the image.

2) Skill Learning: With the parameters estimated in the previous part, we can encode the writing skill through GMR. The normal GMR method can generate the trajectory very well. However, the effect of learning the deviation among samples in the image is not that good. Therefore, we proposed a novel approach called deviation analysis to enhance the deviation learning ability of GMR.

3) Motion Generalization: It is common to adjust the generalization effect of writing skills in different situations, like changing the size of the written font to adapt the various types of papers in the real world. DMP is used to make the skill have great generalization performance. DMP can encode not only the trajectory but also the end-effector orientation.
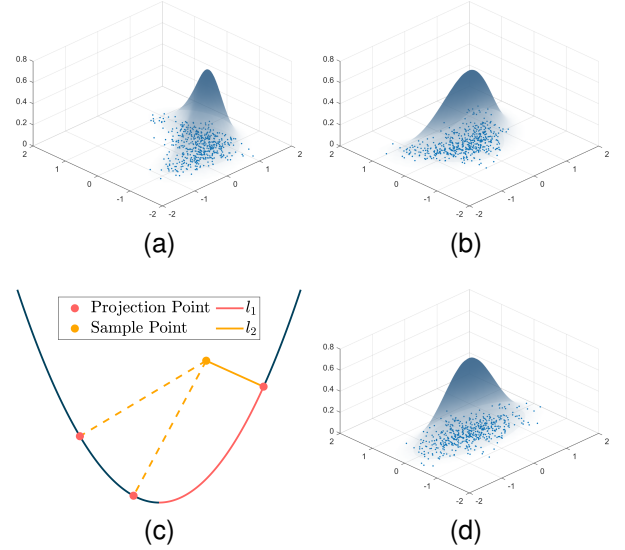

Fig. 2. The data transformation process of FGMM.

B. Fuzzy Gaussian Mixture Model

Gaussian Mixture Models have been widely applied in various fitting scenarios. However, due to the linearity of the axes in GMM, more components are required to fit nonlinear data. Paper [1] introduce a modified GMM called Fuzzy Gaussian Mixture Model (FGMM). It combines the conventional GMM with Active curve axis Gaussian mixture models (AcaG) [2] and introduces a dissimilarity function to accelerate convergence. Fig. 2 represents the data transformation process of FGMM.

$X = \{x_1, \dots, x_N\}$ is a d-dimensional observed dataset

with N vectors. Normally, we assumed that the distribution of X is based on the mixture Gaussian distribution, which the following equation can represent:

$$p(x_n|\mu, \Sigma) = \sum_{i=1}^{K} \alpha_i p_i(x_n|\mu_i, \Sigma_i) \qquad (1)$$

where x_n is a d-dimensional vector and α_i, μ_i, Σ_i mean the ith Gaussian component's weight, mean and variance respectively. K is a pre-set parameter representing the number of Gaussian components. However, the data with nonlinear characteristics do not fit this assumption well, like the data presented in Fig.2a. AcaGMM introduces non-linear factors to the conventional GMM. In the following part, we introduce the definition and parameter estimation process of AcaGMM.

First, the k-means algorithm divides the dataset into K classes. For each class of data, we transfer them to a new coordinate through PCA (Principal Component Analysis) :

$$Y_i = Q_i \times (X_i - T_i) \qquad (2)$$

where Q_i and T_i denote the rotation matrix and translation vectors. X_i is data which belongs to the ith components. The transformed data Y_i presented in Fig.2b have zero mean, and their principal axis aligns with the transformed x-axis. Noticed that we assume all data are two-dimensional in this paper. The LSFM (least-squares fitting method) is used to fit the data using a parabolic curve, which is defined as:

$$y = cx^2 + b \qquad (3)$$

where c denotes the curvature of data and b is the bias. Assuming all data satisfies the parabolic equation, the linear function can be written as:

$$\begin{bmatrix} (y_1)^2 \\ (y_2)^2 \\ \vdots \\ (y_n)^2 \end{bmatrix} = \begin{bmatrix} (x_1)^2 & 1 \\ (x_2)^2 & 1 \\ \vdots & \vdots \\ (x_n)^2 & 1 \end{bmatrix} \begin{bmatrix} c \\ b \end{bmatrix} \qquad (4)$$

where the equation can be written as $Y = XC_i$. The least-squares solution is $C_i = (X^T X)^\dagger X^T Y$, where $\dagger$ means Moore–Penrose pseudoinverse. For components with small values of c, they can be degenerated into a conventional GMM:

$$y = \begin{cases} b & (|c| < \epsilon) \\ cx^2 + b & (|c| > \epsilon) \end{cases} \qquad (5)$$

where ϵ is a small positive number. If $|c|$ is smaller than ϵ, the principal axis can be approximated as a straight line, and the component can be seen as a conventional GMM component. Otherwise, if $|c|$ is bigger than ϵ, the component can be considered as an AcaGMM component.

AcaGMM uses the projection point on the parabolic curve to replace the sample point for calculating the posterior probability. First, the distance between a sample point $\{x_{1t}, x_{2t}\}$ and its projection point $\{z_1, z_2\}$ satisfies:

$$\begin{aligned} d^2 &= (z_1 - x_{1n})^2 + (z_2 - x_{2n})^2 \\ z_2 &= cz_1^2 + b \end{aligned} \qquad (6)$$

The Lagrange method is used to find the closest projection point:

$$2c^2 z_1^3 + (2bc - 2cx_{2n} + 1)z_1 - x_{1n} = 0 \qquad (7)$$

the cubic function may have multiple solutions, which means a sample may have multiple projection points with the shortest distance. Then we calculate the arc length of the jth projection points $\{z_{1j}, z_{2j}\}$:

$$l_{1j} = \int_{(0,b)}^{(z_{1j}, z_{2j})} \sqrt{(dz_{1j})^2 + (dz_{2j})^2} \qquad (8)$$

eliminate z_{2j}, we have:

$$l_{1j} = \frac{1}{2} z_{1j} \sqrt{4c^2 z_{1j}^2 + 1} + \frac{1}{4|c|} ln(2|c|z_{1j} + \sqrt{4c^2 z_{1j}^2 + 1}) \quad (9)$$

and the distance between x_t and z_j is:

$$l_{2j} = \sqrt{(x_{1t} - z_{1j})^2 + (x_{2t} - z_{2j})^2} \qquad (10)$$

where fig 2c illustrates the geometric meaning of l_{1j} and l_{2j}.

Then we can calculate the probability of the sample x_t via the sum of its projection points z_j:

$$p_i(x_n|\theta_i) = \sum_{j=1}^{J_t} \frac{exp(-l_{1j}^2/2\Sigma_1 - l_{2j}^2/2\Sigma_2)}{2\pi\sqrt{|\Sigma_1\Sigma_2|}} \qquad (11)$$

where the parameter $\theta_i = (\mu_i, \Sigma_i, C_i, Q_i, T_i)$, and J_t means the number of projection points of the sample x_n.

The EM algorithm is proposed for parameter estimation in AcaGMM. Due to the change in the probability density function, the parameter estimation method of AcaGMM differs from the conventional GMM. The procedure of the new EM algorithm is:

1) E-step: For each component, calculate the "expected" components of all samples:

$$w_{in} = \frac{\alpha_i p_i(x_n|\theta_i)}{\sum_{s=1}^{k} \alpha_s p_s(x_n|\theta_s)} \qquad (12)$$

where w_{in} denotes x_n posterior probability for the ith components.

2) M-step: For each component, calculate the maximum likelihood of the given sample. The weight of each component is then estimated from:

$$\alpha_i = \frac{1}{N} \sum_{n=1}^{N} w_{in} \qquad (13)$$

Then parameters C_i, T_i, Q_i can be estimated from:

$$(C_i^{new}, T_i^{new}, Q_i^{new}) = LSFM(PCA(X_i)) \qquad (14)$$

where X_i denotes samples belonging to the i_{th} components, which can be classified by Eq. 12. μ_i can be estimated by the following method:

$$\mu_i^{new} = \frac{\sum_{n=1}^{N} w_{in} x_n}{\sum_{n=1}^{N} w_{in}} + (Q_i^{new})^T [0, b]^T + T_i^{new} \qquad (15)$$

and Σ_i can be estimated from:

$$L_{sn} = \frac{\sum_{j=1}^{J_{in}} p(l_{sj}^i|0, \Sigma_{si}^{old}) l_{sj}^i}{\sum_{j=1}^{J_{in}} p(l_{sj}^i|0, \Sigma_{si}^{old})} \quad (s = 1, 2) \qquad (16)$$
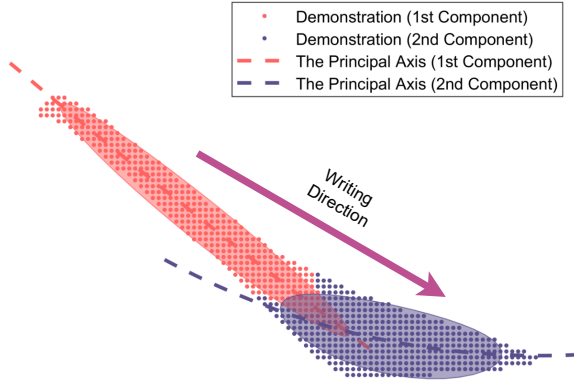
Fig. 3. Fitting Chinese stroke using FGMM.

Fig. 4. The result of time encoding and GMR.

$$\Sigma_{si} = \frac{\sum_{n=1}^{N} w_{in} L_{sn}}{\sum_{n=1}^{N} w_{in}} \ (s = 1, 2) \tag{17}$$

where L_{1n} denotes the mean arc length of x_n's projection points, and L_{2n} means the average distance between x_n and its projection points. It can be considered that samples X_n undergoes a non-linear transformation to obtain $L_n = \{L_{1n}, L_{2n}\}$, presented in Fig.2d. L_n satisfied the distribution of the conventional zero-mean Gaussian.

The above step represents a single iteration of the EM algorithm. Repeat the iteration and update the parameters until the relative difference of log-likelihood between two iterations is less than a preset threshold. FGMM introduces an FCMs (Fuzzy C-means) mechanism to accelerate the convergence speed of the EM algorithm.

C. Time Encoding

Traditional vision-based demonstrations are mostly obtained by recording motion sequences to acquire demonstration trajectories. Data usually contains time sequences, and time-driven methods like GMR and DMP are used for motion encoding and skill learning. However, in some scenarios, the time sequence of demonstration trajectories is missing, like demonstrating writing skills through an image. A new method based on FGMM is proposed for encoding the time series of demonstration trajectories.

First, estimate the parameter of FGMM using the EM algorithm proposed before. Here we use a Chinese character stroke image as an example, presented in Fig.3. The direction is needed to determine a general direction of movement. The direction does not need to be very precise. Most Chinese character strokes follow a sequence from left to right and from top to bottom, so the order of FGMM components can be determined. Each sample has a transformed point L_n after a non-linear transformation during the parameter estimation. For most writing tasks involving movement along the principal axis, time information can be obtained through L_{1n}, which denotes the relative position of the sample on the principal axis. First we transfer the transformed sample from the first component $L_1^{(1)} = \{L_{11}^{(1)}, \ldots, L_{1n}^{(1)}\}$ to a zero starting point:

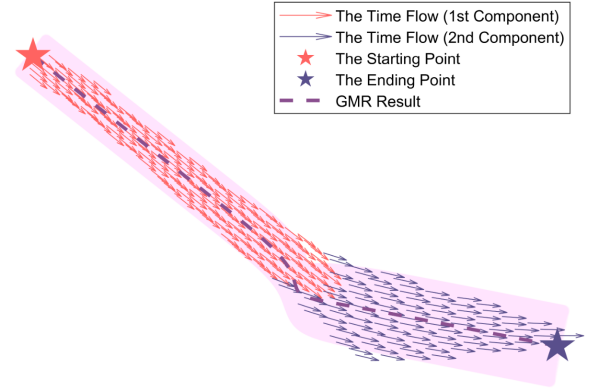$$t^{(1)} = \{L_{11}^{(1)}, \ldots, L_{1n}^{(1)}\} - min(L_1^{(1)}) \tag{18}$$

For the second component of transformed sample $L_1^{(2)}$, transfer them to the end of the first sequence:

$$t^{(2)} = \{L_{11}^{(2)}, \ldots, L_{1n}^{(2)}\} - min(L_1^{(2)}) + max(L_1^{(1)}) \tag{19}$$

if there are multiple components, repeat this process until the end. The time sequence is denoted as $\{t^{(1)}, t^{(2)}, \ldots, t^{(i)}\}$. Fig.4 presents the result of time encoding. The arrow on the sample point indicates the direction of time flow.

After time encoding, the time-driven method can be used for motion planning in writing tasks. Gaussian Mixture Regression (GMR) is a statistical learning technique that combines the GMM with linear regression. GMR is used to address non-linear regression problems by capturing the intrinsic structure of the data through a Gaussian mixture model, which is then used to perform regression analysis. GMR allows us to specify one dimension of the GMM as input, with the remaining dimensions as output. Here, the time sequence t obtained before is used as the input, and the position $X = \{x_1, \ldots, x_n\}$ as the output to plan the writing trajectory. The regression function can be considered as obtaining the expectation of an event for the output X conditional on the input t:

$$\mathcal{F}_{gmr}(t) = E(X|t) \tag{20}$$

Assume that X and t follow the Gaussian distribution, the event satisfies:

$$X|t \sim G(\mu_X + \frac{\sigma_{21}}{\sigma_1}(t - \mu_t), \sigma_2 - \frac{\sigma_{12}\sigma_{21}}{\sigma_1}) \tag{21}$$

where σ_1 and σ_2 denote the variance of t and X, σ_{12} and σ_{21} denote the covariance between t and X. Consider that the Gaussian mixture model can be seen as a linear combination of Gaussian distributions, the expectation can be written as:

$$\mathcal{F}_{gmr}(t) = \sum_{i=1}^{K} w_i(t)\eta_i(t) \tag{22}$$

where

$$w_i(t) = \frac{\alpha_i G(t|\mu_i(t), \sigma_{it})}{\sum_{i=1}^{K} \alpha_i G(t|\mu_i(t), \sigma_{it})} \tag{23}$$

$$\eta_i(t) = \mu_{iX} + \frac{\sigma_{21i}}{\sigma_{1i}}(t - \mu_{it}) \tag{24}$$

In the previous part, the GMM is replaced with FGMM, which has better non-linear fitting capability. The regression

function should be redesigned. Paper [3] proposes a new regression algorithm for FGMM. Notice that Eq.24 is a linear function, it defines the location of GMM's principal axis. For the FGMM, the corresponding axis can be obtained from the PCA inverse transformation:

$$A'_i = Q_i^{\mathbf{T}} A_i + T_i \tag{25}$$

where A_i and A'_i denote the original axis and transformed axis. The regression for FGMM can be denoted as:

$$\mathcal{F}'_{gmr}(t) = \sum_{i=1}^{K} w'_i(t) A'_i \tag{26}$$

where the FGMM component should replace each Gaussian component in w'_i, the movement trajectory planning by GMR is presented in Fig.4. Compared with Fig.3, it can be seen that the trajectory highly overlaps with the principal axis.

D. Deviation Analysis

In some scenarios, such as transferring writing skills to robots, they need to learn not only the trajectory information of sample points but also the deviation among them. The pink area shown in Fig.4 denotes the covariance of trajectory generated by GMR. The fitting effect is not ideal. A new method based on FGMM is proposed. Analyse the deviation of samples to generate a variance sequence for a better fitting effect.

In addition to generating a trajectory from time input, GMR can also generate a corresponding covariance sequence $\{\Sigma_1, \ldots, \Sigma_t\}$. It denotes the covariance among samples in time t. The covariance matrix implicitly contains the posture and deviation of the trajectory. They can be obtained by eigenvalue decomposition:

$$\Sigma_t = R_t D_t R_t^{-1} \tag{27}$$

where D_t and R_t represent the eigenvalues and eigenvectors of the covariance matrix, respectively. According to the geometric meaning of eigenvalue decomposition, they also represent the scaling effect and rotation effect of the covariance matrix, respectively, which represent the deviation and posture information of the trajectory. L_{2n} defines the distance between the transformed sample and the principal axis. It can be used to define a new scaling matrix. According to the 3σ rule, the new scaling matrix is defined as:

$$D_t^{new} = \begin{bmatrix} (\frac{1}{3}L_{2t})^2 & 0 \\ 0 & D_{22t} \end{bmatrix} \tag{28}$$

the new covariance sequence is:

$$\Sigma_t^{new} = R_t D_t^{new} R_t^{-1} \tag{29}$$

E. Dynamic Movement Primitive

To enable the proposed method to adapt to writing tasks in different environments, the DMP is combined into the designed skill learning framework. The DMP is a LfD model based on dynamic systems [4]. It consists of a second-order dynamic system and a non-linear function, allowing it to encode motion trajectories with fewer parameters while

maintaining stability. For the encoded trajectory, modifying the positions of the starting and ending points can also retain the original motion characteristics:

$$\ddot{y} = \alpha(\beta(g - y) - \dot{y}) + f \tag{30}$$

To change the trajectory speed, a scaling term τ is added in front of the velocity variable $\dot{y}$:

$$\tau^2 \ddot{y} = \alpha(\beta(g - y) - \tau \dot{y}) + f \tag{31}$$

To decouple the non-linear equation from time, a canonical system $x(t)$ is used to drive the dynamic system:

$$\tau \dot{x} = -\kappa x \tag{32}$$

The first term in Eq.30 is a PD (Proportional-Derivative) system. To avoid overshooting and ensure stable convergence to the attractor, let the parameter $\beta = \alpha/4$, rewriting the second-order dynamic system as a critically stable spring-damping system. The system ultimately converges stably to the position g. The second term f in Eq.30 is a non-linear function that controls the intermediate process of convergence. The non-linear function f can be expressed as:

$$f(x) = \frac{\sum_{i=1}^{N} \Psi_i(x) w_i}{\sum_{i=1}^{N} \Psi_i(x)} x(t)(g - y) \tag{33}$$

where, $\Psi(t)$ is a Radial Basis Function (RBF), expressed as:

$$\Psi_i(x) = \exp\left(-\frac{(x - \mu_i)^2}{2\sigma_i^2}\right) \tag{34}$$

where N represents the number of RBF, μ_i represents the position of the ith basis function, and σ_i represents the width of the ith basis function. ω_i represents the weight of the ith radial basis function, which needs to be obtained by learning from the demonstration trajectory. Given a demonstration trajectory, here we used the trajectory $y_{dms} = \mathcal{F}'_{gmr}(t)$ obtained by GMR, the expression for f_{dms} can be derived from Eq.30:

$$f_{dms} = \tau^2 \ddot{y}_{dms} - \alpha(\beta(g - y_{dms}) - \tau \dot{y}_{dms}) \tag{35}$$

where Locally Weighted Regression (LWR) is used to determining the weight parameter w_i, which can control the $f(x)$ to fit f_{dms}.

III. EXPERIMENTS

A. Simulation

In this part, The proposed method is verified through simulation. A Chinese calligraphy image is used as the demonstration image, denoted in Fig.5. Paper [5] propose an observation based algorithm which can extract strokes from the Chinese character image. As shown in Fig.5(a), the Chinese character is divided into four strokes. Each stroke contains temporal information, which can be used to indicate the writing direction and the order of seven components, as Fig.5(b) and (c) denotes. Fig.5(b) and (c) also show the result of fitting the Chinese character image by conventional GMM and FGMM, respectively. The conventional GMM need to use
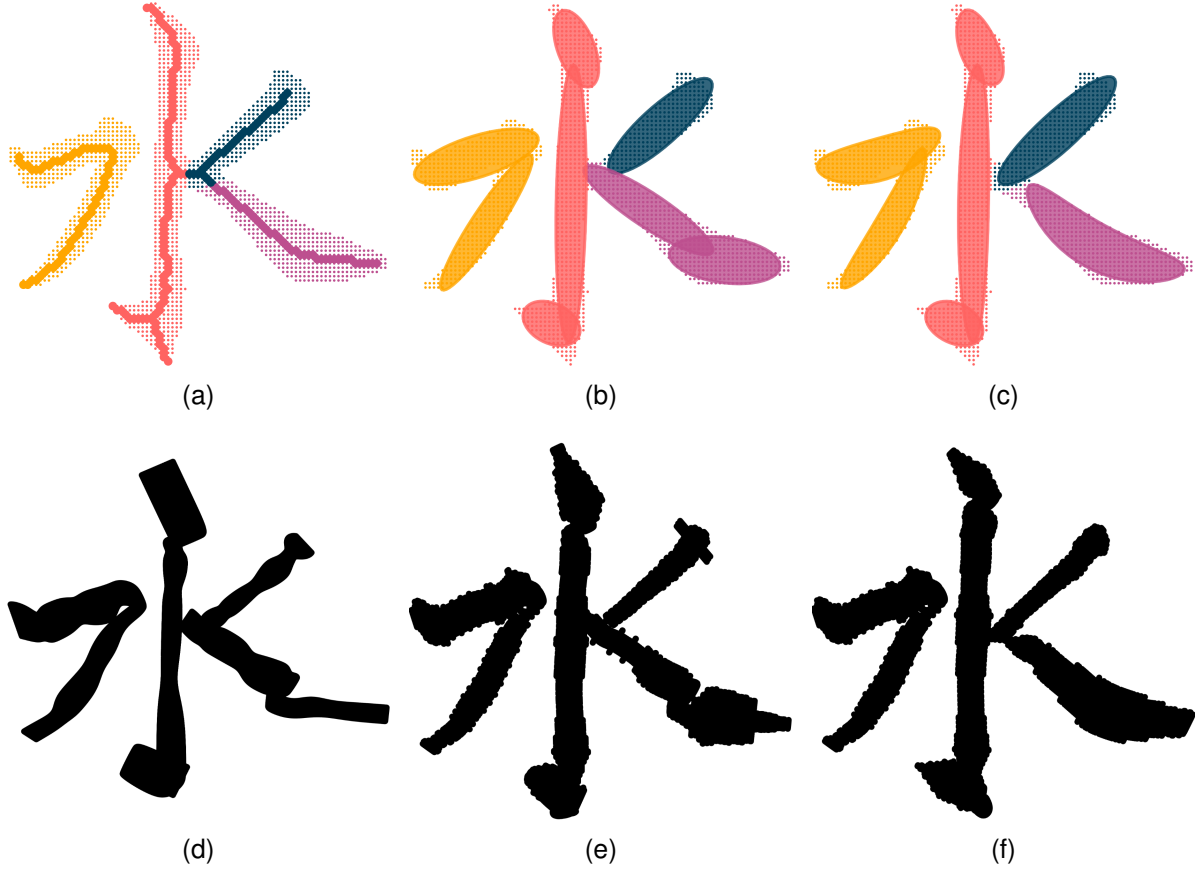
Fig. 5. The writing simulation of Chinese character "water".

8 components to fit the character well. In contrast, the FGMM only requires 7 components.

The sample points are time-encoded for each stroke based on the known writing direction and component order. The new GMR method and DA is used to learn the character's writing trajectory and deviation degree. In the simulation part we need to simulate the effect of robot writing on the computer. The following three methods will be compared to the effect of writing tasks: the GMM without DA, the GMM with DA and the FGMM with DA. To maintain the accuracy of the comparison, the writing tasks are simulated based on a unified method. Assume the discrete trajectory is $\{(x_1, y_1), \ldots, (x_N, y_N)\}$, and the covariance matrix sequence is $\{\Sigma_1, \ldots, \Sigma_N\}$. At time t, the writing trajectory is defined as:

$$o_t = [cos(p), sin(p)]\sqrt{3\Sigma_t} + [x_t, y_t] \quad (-\pi < p < \pi) \quad (36)$$

where the area represented by o_t indicates the writing trace at time t. The writing simulation results for the three methods are denoted in Fig.5.

To compare the fitting effect of the simulation results with the demonstration image, we introduce 10 commonly used Image Quality Assessment (IQA) methods as evaluation metrics, denoted in Table.I. They are all Full Reference (FR) methods, which can evaluate the differences between the simulation results and the demonstration image. The evaluation results are presented in Table.II. The DA improve the fitting effect of

TABLE I
IQA METRICS AND MEANINGS

Metric	Description
PSNR	Peak signal-to-noise ratio. A higher score indicates a higher similarity.
FSIM	Feature Similarity Index. A higher score indicates a higher similarity.
SSIM	Structural Similarity Index. A higher score indicates a higher similarity.
VSI	Visual Saliency-based Index. A higher score indicates a higher similarity.
VIF	Visual Information Fidelity. A higher score indicates a higher similarity.
IFC	Image Fidelity Criterion. A higher score indicates a higher similarity.
DISTS	Differential Stimulation Similarity. A lower score indicates a higher similarity.
MSE	Mean Square Error. A lower score indicates a higher similarity.
VOI	Variation of Information. A lower score indicates a higher similarity.
LPIPS	Learned Perceptual Image Patch Similarity. A lower score indicates a higher similarity.

Chinese character a lot. The improvements in FGMM further enhance the fitting effect.

B. Generalization Verification

In this part we will verify the generalization performance of the method. In the real world, the writing skills should be
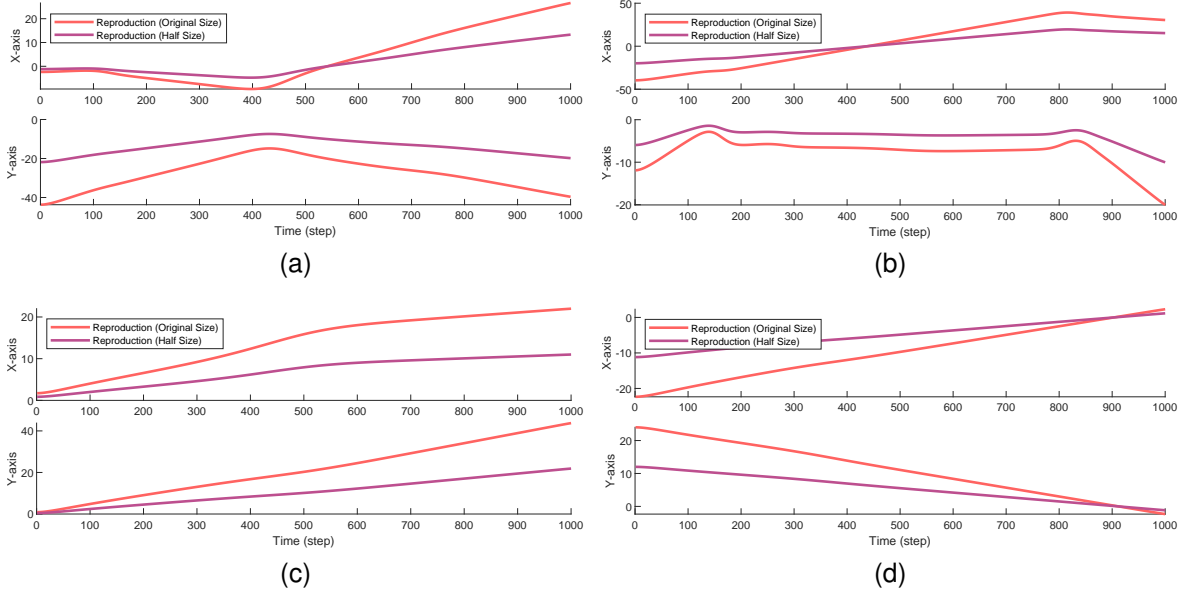
Fig. 6. The reproduction of writing skill for each stroke.

TABLE II
THE IQA RESULT OF THE SIMULATION

Metric	GMM	GMM+DA	FGMM+DA
PSNR↑	10.7447	11.5709	**12.1143**
FSIM↑	0.7896	0.7901	**0.7990**
SSIM↑	0.7867	0.7928	**0.8019**
VSI↑	0.7665	0.8018	**0.8126**
VIF↑	0.0203	0.0225	**0.0236**
IFC↑	0.0452	0.0493	**0.0514**
VOI↓	2.9374	2.8622	**2.8019**
MSE↓	5477.8718	4528.9159	**3996.2290**
DISTS↓	0.3535	0.3522	**0.3463**
LPIPS↓	0.3354	0.3058	**0.2951**

able to reproduce correspondingly in a variety of scenarios, like writing in different sizes of paper. Here the same Chinese character is used as an example to verify the performance of the writing skills reproduction.

Assuming the paper size for reproducing the writing skills is half the size of the demonstration image, the starting and ending points of the Chinese character strokes need to be re-determined. Here, the one-point perspective is used to determine the new starting and ending points. Then the proposed framework can reproduce the writing skills to fit this new scenario. Fig.6 denotes the demonstrative samples after time encoding and the reproduction of writing skills in X and Y dimension. The reproduction trajectories and their starting and ending points are represented in Fig.7.
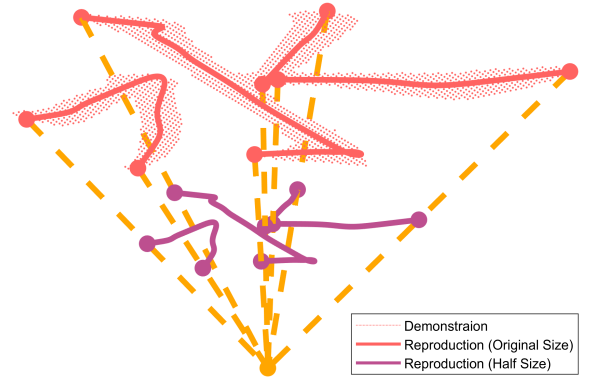

Fig. 7. The new starting and ending points for each stroke determined by one-point perspective. And the reproduction of writing skill for the whole Chinese character.

[3] C. Yang, C. Chen, N. Wang, Z. Ju, J. Fu, and M. Wang, "Biologically inspired motion modeling and neural control for robot learning from demonstrations," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 11, no. 2, pp. 281–291, 2019.
[4] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.
[5] X. Li, W. Si, and C. Yang, "An observation based method for human robot writing skill transfer," in *2022 IEEE 17th International Conference on Control & Automation (ICCA)*, 2022, pp. 412–417.

C. Real-World Experiment

IV. CONCLUSION

REFERENCES

[1] Z. Ju and H. Liu, "Fuzzy gaussian mixture models," *Pattern Recognition*, vol. 45, no. 3, pp. 1146–1158, 2012.
[2] B. Zhang, C. Zhang, and X. Yi, "Active curve axis gaussian mixture models," *Pattern Recognition*, vol. 38, no. 12, pp. 2351–2362, 2005.