Learning to Reproduce Visually Similar Movements by Minimizing Event-Based Prediction Error

Conference Paper · October 2018

DOI: 10.1109/BIOROB.2018.8487959

6 authors, including:

Jacques Kaiser
FZI Forschungszentrum Informatik
34 PUBLICATIONS **314** CITATIONS
SEE PROFILE

J. Camilo Vasquez Tieck
FZI Forschungszentrum Informatik
28 PUBLICATIONS **200** CITATIONS
SEE PROFILE

Arne Roennau
FZI Forschungszentrum Informatik
127 PUBLICATIONS **861** CITATIONS
SEE PROFILE

Martin Butz
University of Tuebingen
302 PUBLICATIONS **4,792** CITATIONS
SEE PROFILE

Learning to Reproduce Visually Similar Movements by Minimizing Event-Based Prediction Error

Jacques Kaiser[1], Svenja Melbaum[1], J. Camilo Vasquez Tieck[1], Arne Roennau[1], Martin V. Butz[2], Rüdiger Dillmann[1]

***Abstract*— Prediction is believed to play an important role in the human brain. However, it is still unclear how predictions are used in the process of learning new movements. In this paper, we present a method to learn movements from visual prediction. The method consists of two phases: learning a visual prediction model for a given movement, then minimizing the visual prediction error. The visual prediction model is learned from a single demonstration of the movement where only visual input is sensed. Unlike previous work, we represent visual information with event streams as provided by a Dynamic Vision Sensor. This allows us to only process changes in the environment instead of complete snapshots using spiking neural networks. By minimizing the prediction error, movements visually similar to the demonstration are learned. We evaluate our method by learning simple movements from human demonstrations on different simulated robots. We show that the definition of the visual prediction error greatly impacts movements learned by our method.**

I. INTRODUCTION

The brain is able to solve difficult control problems through learning. If we were to understand how the brain learns, robots would be able to master their body like humans do, without being assigned specific goals. While we are still far from understanding how the brain learns, it is believed that prediction plays an important role [1–7]. Indeed, the brain is capable of predicting future sensory input originating from its own actions (forward model), as well as estimating what action it should take to reach a specific sensory input (inverse model) [8–10]. It has also been stated that the brain learns new forward models faster than associated inverse models [11]. Based on these insights, we introduce a method to learn new movement event models, as characterized in [4], from a visual prediction model. Short-term visual predictions are learned from event streams in a biologically inspired manner with spiking neural networks. The event streams used for learning predictions are acquired from demonstration. This makes the evaluation of our method simpler, as the learned movements can be qualitatively compared to the demonstrated ones. However, one could train a visual prediction model without demonstration, in an unsupervised learning fashion [8, 12]. To learn a movement from a visual prediction model, we use an optimization where the robot searches for the movement that is the most visually similar to the predicted one. An overview of our method is provided in Fig. 1. Since our method does not require a mapping from the demonstrator to the platform, a robot can learn new movements from human demonstrations. However, demonstrations have to be provided in first-person view.

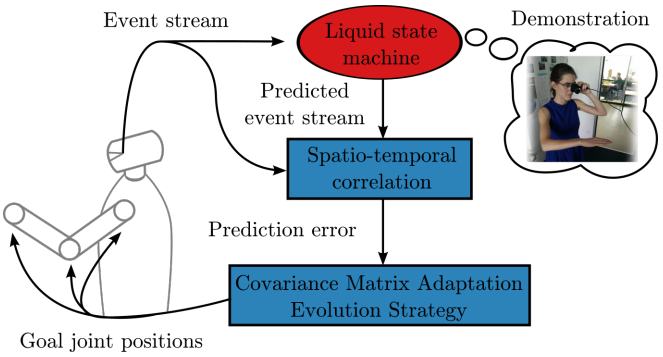

Fig. 1: Our method to learn movements by imitation from event-based visual prediction. First, a visual predictive model is learned from a demonstration with a liquid state machine (see Fig. 2), as presented in [13]. A single demonstration is required, but they should be provided in first-person view. Second, the robot tries different movements iteratively in an attempt to minimize the prediction error. We rely on covariance matrix adaptation evolution strategy (CMA-ES) [14] to generate goal joint positions. The robot performs a movement by controlling its joints from the initial positions to the goal positions.

The main contribution of this paper is the novel proposed method which provides another view on how robots can learn to move with few data and little modeling. Since visual information is represented with event streams, only motion is sensed and we circumvent the complicated problem of predicting frames. This is also valid with respect to biology, as biological retinas sense changes in the environment and adapt rapidly to static visual stimuli [15]. Moreover, it is believed that motion information is processed separately from color and shape in the visual cortex, and is especially used for motor control [16]. In addition, we show that the definition of the visual similarity between event streams greatly impacts movements learned by our method. Specifically, the robot stays motionless with the prediction error definition proposed in [13], based on absolute difference (see Fig. 6). Instead, we propose to compute prediction error using the Pearson correlation, enabling movements to be learned. While our

[1]FZI Research Center for Information Technology, 76131 Karlsruhe, Germany. {jkaiser,melbaum,tieck,roennau,dillmann}@fzi.de [2]Cognitive Modeling, University of Tübingen, Germany. martin.butz@uni-tuebingen.de

method focuses on reproducing seen movements, we discuss how it could be extended to learn goal-directed behavior in an online fashion in Section V. The presented method is an effort to bring neuroscience and robotics closer together by experimenting insights from neuroscience on robotic platforms. This paper is an extension of the abstract first published in [17].

After a literature review on learning from visual prediction in Section II, our method is introduced in Section III. Specifically, we present our method to learn visual prediction from event stream in Section III-A, our definition of the visual prediction error in Section III-B, and our optimization routine in Section III-C. Our method is evaluated in Section IV with human and simulated robot demonstrations. We conclude in Section V.

II. Related Work

Recently, many methods have been presented to predict future visual input [18–20]. However, most of the previous works only consider video data represented as a sequence of frames. It has been stated that this representation was optimal for movies and art, but not adequate for extracting information, as needed in artificial vision [21]. Meanwhile, event-based methods based on reservoir computing to predict asynchronous visual input have been presented in [13, 22, 23]. In this paper, we rely on [13] to learn event-based visual prediction models from event streams, see Section III-A for more details.

Minimizing prediction errors has been shown to yield promising behaviors in artificial intelligence and robotics [9, 24–26]. In [9], new body-specific behaviors are learned solely from prediction error. Behaviors are represented as a combination of forward (predictive) models and inverse (controller) models. This method was successfully applied to control tendon-driven robots [27]. However, the input space only consists of the proprioception for all joints.

In this paper, we learn a visual prediction model from demonstration, and the associated inverse model by minimizing the visual prediction error. The closest works to our method are [25, 26]. Similar to our method, a robot learns new movements by minimizing visual prediction error with respect to a demonstration.

In [25], a robot learns to interact with humans (clean table when human approaches, push trivet when human comes with pan) in real-time by combined use of trained neural networks. The method consists of two components: perception and manipulation. The perception component is trained to predict future hand locations from human first-person video input. The manipulation component is trained to map robot hand locations, joint states and predicted hand locations to future joint states. At test time, the robot is able to move its hand to predicted hand locations. This work requires a pre-trained convolutional neural network (CNN) for hand detection. Compared to [25], our method is more generic and does not rely on pre-trained networks solving specific tasks.

In [26], simple object manipulation in a table-top environment is learned. In this work, separate predictive networks are trained for different tasks on the demonstrator side and different discrete actions on the agent side. At test time, the action minimizing the difference between predicted state on demonstrator and agent side is chosen. Unlike [26], we use a continuous action space - the robot can drive to any goal joint configuration.

Similar to [13], our method innovates with the use of event-based vision and spiking neural networks. By using an event-based vision sensor instead of a classical camera, only changes in the environment are sensed, while the environment's state is stored in the activity of a recurrent spiking neural network. By using an event-based representation of the visual input, we circumvent the challenge of predicting complete frames by only predicting pixel changes. However, since we use a different visual representation, a new visual similarity measure needs to be defined (see Section III-B).

III. Imitation learning from prediction

We show how minimizing the visual prediction error can be used to learn by imitation. In this setup, a robot is able to learn new movements by watching a teacher's demonstration in first-person view. During the demonstration, only the visual sensory input is recorded in form of an event stream. No mapping is needed between the learner and the teacher.

An overview of our method is presented in Fig. 1. In the first phase, we train a spiking neural network that can predict future visual input from a single demonstration. This prediction model represents the memory of the demonstrated movement. In the second phase, the robot tries to find the movement most visually similar to the predictable one. The similarity between predicted and performed movements is evaluated from the visual prediction error. By minimizing the visual prediction error, the method recovers a movement visually similar to the demonstration.

A. Event-based predictions

We rely on the method presented in [13] to compute short-term visual prediction from an event-based sensor. Predicting future pixel values from a camera is known to be a hard problem [24]. By using an event-based sensor, we circumvent this challenge by only predicting changes in pixel intensity.

The event-based predictions are provided by a liquid state machine trained on an event stream, see Fig. 2. A liquid state machine is a recurrently connected spiking neural network, where only on the readout weights are learned with a linear regression [28]. Unlike classical analog neurons, spiking neurons model precise spike time, and are therefore suited to asynchronous processing [29].

For simplicity, we reuse the mathematical notations introduced in [13], where vectors are denoted with bold lowercase letters and matrices with bold uppercase letters. The event stream is fed in recurrently connected spiking neurons, called the liquid. A subset of n_{rec} excitatory liquid neurons are connected to each readout neuron in the output layer, which has the same dimensionality as the input. The activity

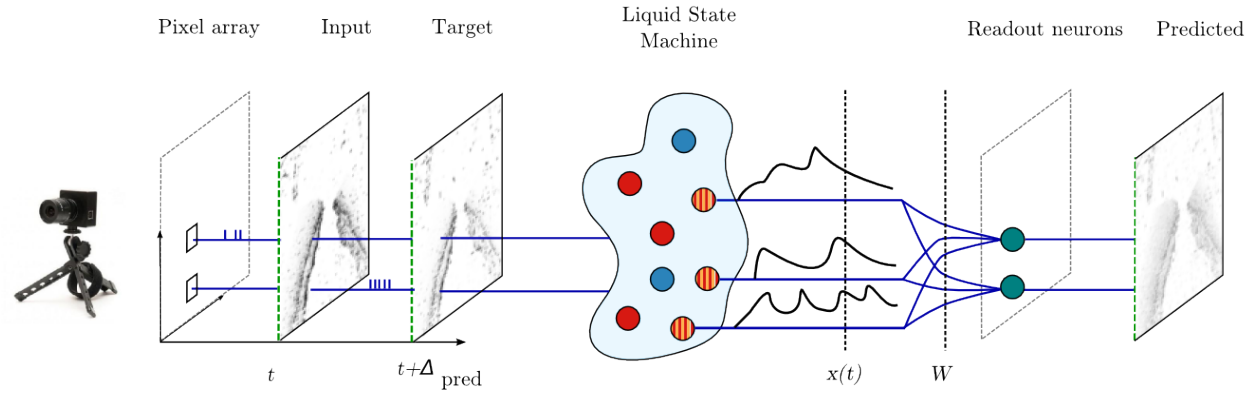

Fig. 2: Liquid state machine used to predict future visual input from address events provided by a Dynamic Vision Sensor (DVS – picture from iniVation AG). The events (ON and OFF) are fed to randomly connected excitatory (●) and inhibitory (●) liquid neurons as spikes. We record the activity of n_{rec} excitatory neurons (●) which we define as the liquid state $\boldsymbol{x}(t)$ for a given time t. Learning consists of finding the weights $\boldsymbol{W}$ mapping liquid states to the target signals. Since we learn a predictive model, the target signals are obtained by applying an exponential filter on the input event stream time-shifted by Δ_{pred}. Image inspired by [13].

relayed by these connections is called the liquid state, and is denoted as $\boldsymbol{x}(t) \in \mathbb{R}^{n_{rec}}$ for a given time t. Only these connections are trained. They are parametrized with the weights $\boldsymbol{W} = [\boldsymbol{w_1}, \ldots, \boldsymbol{w_p}] \in \mathbb{R}^{n_{rec} \times p}$, with p the number of readout neurons. The training consists of a simple supervised linear regression from the liquid states to the target signals, defined as:

$$\boldsymbol{X} \cdot \boldsymbol{W} = \boldsymbol{B}, \tag{1}$$

with

$$\boldsymbol{X} = \begin{bmatrix} | & | & & | \\ \boldsymbol{x}(t_1) & \boldsymbol{x}(t_2) & \ldots & \boldsymbol{x}(t_{n_{samples}}) \\ | & | & & | \end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^{n_{samples} \times n_{rec}}$$

the accumulated sampled activities, and

$$\boldsymbol{B} = \begin{bmatrix} | & | & & | \\ \boldsymbol{b}(t_1) & \boldsymbol{b}(t_2) & \ldots & \boldsymbol{b}(t_{n_{samples}}) \\ | & | & & | \end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^{n_{samples} \times p}$$

the sampled target signals for all p readout neurons. The target signals are obtained by applying an exponential filter on the input event stream time-shifted by Δ_{pred}:

$$\boldsymbol{b}(t) = \begin{bmatrix} | \\ exp\left(-\frac{t + \Delta_{pred} - t_i^{spike}}{\tau}\right) \\ | \end{bmatrix},$$

where $t_i^{spike} \in]-\infty, t]$ denotes the last spike time of input neuron i, and τ a global fading term. At a given timestep t, we can denote the visual prediction $\boldsymbol{p}(t)$ as:

$$\boldsymbol{p}(t) = \boldsymbol{x}(t)^{\mathsf{T}} \cdot \boldsymbol{W}. \tag{2}$$

As noted in [13], only a single training motion is sufficient to provide predictions for similar motions, when the scene is not crowded by many moving objects. For our approach, this means that one single demonstration is enough for the apprentice to learn a new movement.

B. Definition of the prediction error

In this paper, we learn movements by minimizing the prediction error with respect to the demonstration. Therefore, the definition of the prediction error greatly impacts learned movements. In [13], the prediction error was computed as the absolute difference between the prediction and the actual movement. This metric was appropriate for evaluating the quality of the prediction. However, this metric admits a major flaw for our method: no movement results in a very low error. Indeed, when there is no movement, both the visual prediction as well as what is perceived by the robot are null (no activity). In this case, the absolute difference between the prediction and the actual movement will also be null. In other words, the null movement is always a global minimum with respect to this metric, for any demonstration. Therefore, minimizing the visual prediction error with this metric will often result in no movement at all, independently of the demonstration. The difference is experimentally illustrated in Section IV.

In this paper, we propose a different definition of the prediction error which does not suffer from this problem. We rely on the Pearson correlation, which measures the linear dependency between two variables. In our case, we calculate visual prediction error for a single prediction as the negative correlation of the predicted and actual activations $\rho(t)$ at a given time t as:

$$\rho(t) = -\frac{\mathrm{cov}(\boldsymbol{p}(t), \boldsymbol{b}(t))}{\sqrt{\mathrm{var}(\boldsymbol{p}(t))\mathrm{var}(\boldsymbol{b}(t))}}, \tag{3}$$

with cov and var the covariance and variance of random variables, respectively. The visual prediction error for a

time sequence is computed as the average error over every timestep[1]:

$$e = \tanh\left(\frac{1}{n_{samples}^{test}} \cdot \sum_{t \in t^{test}} \operatorname{arctanh}(\rho(t))\right). \quad (4)$$

C. Minimizing the prediction error

The prediction error measures the similarity between the predicted movement and the performed movement. By minimizing the prediction error, we can recover movements visually similar to demonstrated movements.

We rely on Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [14] to minimize the prediction error. CMA-ES combines a standard (μ, λ)-evolution strategy (a weighted combination of the μ best out of λ offspring survive an iteration), with an elaborated covariance matrix update. At each iteration, offspring are generated by sampling a multivariate Gaussian distribution. The mean and the covariance of the distribution are updated with respect to the fitness function evaluated for each offspring. This method has the advantage of handling the decrease of exploration throughout the iterations, and is therefore easy to employ, since only the initial solution μ and an initial variance σ have to be chosen. In our case, each offspring represents desired goal joint positions. For each offspring, the robot controls its joints from the initial positions to the goal positions. We initialize CMA-ES with a null movement, i.e. the initial solution μ is the initial joint positions. At a given iteration step, offspring (vectors of goal joint positions) are generated and performed by the learner. For each offspring, we evaluate the visual prediction error as defined in Equation 4 and use it as a fitness function. We terminate learning after a given number of iterations, when CMA-ES converges to goal joint positions resembling the demonstrated movement.

IV. Evaluation

In this section, we evaluate the ability of our method to learn movements from a visual demonstration. We perform three experiments. In the first experiment, we validate the visual correlation error metric defined in Equation 4 against a classical absolute error as proposed in [13]. In the second experiment, we show that the simulated robotics platform can learn from a human demonstrator. In the third experiment, we evaluate whether the method can learn multiple movements from the same demonstration, with respect to a visual cue. The three experiments are detailed in the next section. The experiments were run in the Neurorobotics Platform[2][30] using the Virtual Coach, which allows for batch optimization.

A. Experimental setup

For each experiment, we first learn a visual prediction model from a demonstration with a liquid state machine. The liquid state machine[3] consists of 1000 excitatory neurons, out of which $n_{rec} = 500$ of them are recorded, and 250 inhibitory neurons. The liquid is trained to predict $\Delta_{pred} = 200ms$ in the future. The inner parameters of the liquid defining its dynamic are taken from [13].

Demonstrations are recorded either in reality with a Dynamic Vision Sensor (DVS) [31] using the ROS interface[4] developed in [32], or in simulation using the gazebo DVS plugin[5] presented in [33]. Both ON and OFF events are sent indistinctly as input spikes. After the visual prediction model is learned, we start the optimization process to learn a movement visually similar to the predictable one. We initialize a robot with similar initial joint positions as in the demonstration. We set the initial solution μ of CMA-ES to the initial joint positions, and the initial variance σ to 0.5 radians per joint. For each iteration, 15 offspring are generated (only 10 for the third experiment). For each offspring of goal joint positions, a movement is executed. During the execution, the perceived event stream is recorded, and we assign the prediction error as fitness value for this offspring. At the end of the execution, the robot goes back to its initial joint positions until another offspring is generated. We terminate the learning after a given number of iterations.

The first experiment is a proof of concept where a simulated schunk arm LWA 4P learns to move its arm to a pose demonstrated in simulation. The simulated schunk arm LWA 4P is fixed to a table and the simulated DVS looks towards the arm. Three joints are used to perform the movement: shoulder, elbow and hand. The movement is demonstrated by the same robot in the same setup, and is perpendicular to the camera view, see Fig. 3.

For the second experiment, we show that the method is also able to learn from a human demonstration and for more joints. In this experiment, a simulated iCub robot learns to move its arm closer together, see Fig. 4. The simulated DVS replaces the left eye of the iCub, looking down to its arms. Twelve joints are used to perform the motion - for each arm: elbow, wrist prosup, shoulder pitch, shoulder yaw, shoulder roll and wrist pitch.

For the third experiment, we evaluate whether the method can learn multiple movements from a single demonstration depending on some visual cue. The demonstration consists of a simulated schunk arm LWA 4P moving to the left when a given visual cue is active, and moving to the right when another visual cue is active. The visual cues are flashing balls on the left or on the right of the image, triggering address events in the simulated DVS, see Fig. 5.

B. Results

For each experiment, we plot learned joint goal positions over iterations of CMA-ES. In the first and third experiments, we additionally plot the ground truth joint goal positions,

[1]The mathematically correct way of averaging a sequence of correlation coefficients is to take the Fisher z-transformation, calculate the mean, and transform back.

[2]https://bitbucket.org/hbpneurorobotics/neurorobotics-platform

[3]https://github.com/HBPNeurorobotics/LSM

[4]https://github.com/uzh-rpg/rpg_dvs_ros

[5]https://github.com/HBPNeurorobotics/gazebo_dvs_plugin

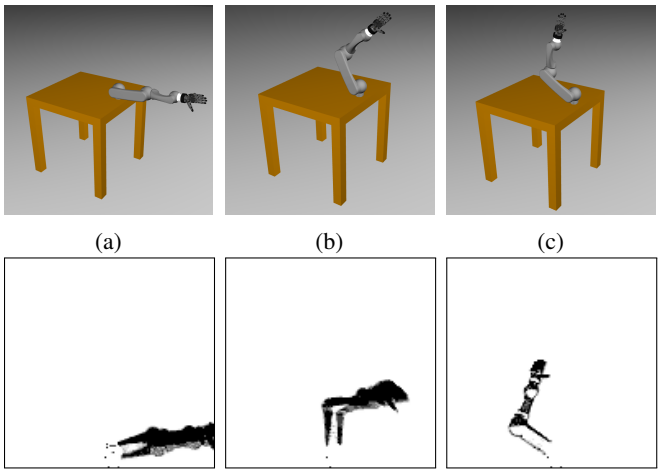

Fig. 3: Visualization of the demonstration for the first experiment. The movement consists of a simulated schunk arm LWA 4P extended straight to the right and moving to the left. First row: third-person view at the beginning (a), during (b) and at the end (c) of the demonstrated motion. Second row: aggregated DVS address events at the beginning (d), during (e) and at the end (f) of the demonstrated motion.

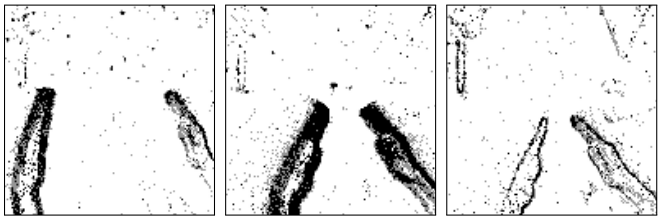

Fig. 4: Visualization of the demonstration for the second experiment. The human demonstrator has a real DVS strapped on his front. The movement consists of bringing the two arms closer together. The images are rendered by aggregating DVS address events at the beginning (a), during (b) and at the end (c) of the demonstration.

which are known as the demonstration was given in simulation with the same robot. For the second experiments, the ground truth is not known as the movement is demonstrated by a human. In this case, we evaluate our method by assessing the visual similarity to the demonstration empirically.

We run the first experiment two times with two different fitness functions (Fig. 6). The first fitness function is the absolute difference between the prediction and the actual input, as proposed in [13]. As seen in Fig. 6a, goal joint positions converge to the initial joint positions positions after 20 iterations. The robot learned to stay motionless (Fig. 3a) with high confidence, as witnesses the absolute difference error metric close to optimal zero (Fig. 6b). Indeed, as mentioned in Section III-B, using the absolute difference as a metric has the pitfall that when there is no motion, both the prediction and and the actual input are void, yielding the

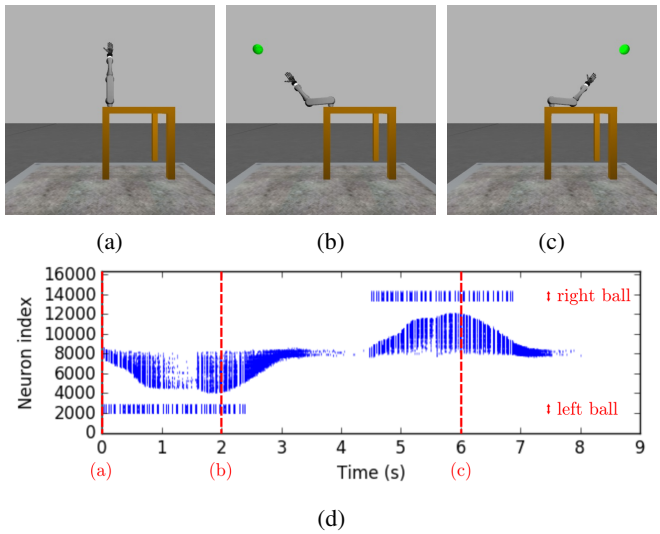

Fig. 5: Visualization of the demonstration for the third experiment. A simulated schunk arm LWA 4P moves left when a ball is flashing on the left, returns to straight position, moves right when a ball is flashing on the right and returns to straight position. First row: third-person view of the demonstrated motion at 0 seconds (a), 2 seconds (b) and 6 seconds (c). (d): Input spiketrain of the demonstration.

absolute difference to be null. In other words, when using the absolute difference as a fitness function, not moving is always a global minimum.

The second fitness function is the negative Pearson correlation between the prediction and the actual input. In this case, no movement and no prediction have a low correlation, hence a high error. We can see that after 12 iterations, precise joint goal positions are recovered by our method (Fig. 6c). This means that the robot learns to move exactly as in the demonstration (Fig. 3c). Again, the decrease in error over iterations is stable, as seen in Fig. 6d, indicating a confident learning process.

Since the second experiment is demonstrated by a human with a real DVS camera, the ground truth joint positions is not known and the learned movement can only be assessed empirically. The learned movement after 50 iterations is shown in Fig. 7, and the learning process in Fig. 8. Like in the demonstration, both arms move symmetrically, without any constraint encouraging symmetry (Fig. 7c). Despite different arm and hand shapes compared to the human demonstrator, the motion perceived by the simulated DVS for the learned movement is similar than the real DVS during demonstration (see Fig. 4 and Fig. 7). As in the first experiment, the stable decrease in error (Fig. 8d) and rapid convergence of goal joint positions after 30 iterations (Fig. 8a and Fig. 8b) show that the method learns with confidence even for 12 joints. Additionally, as the simulated iCub manages to learn a similar movement to the demonstrated one, we can conclude that the visual prediction model learned with the real DVS can also be used for the simulated DVS, despite the inaccurate simulation [33].

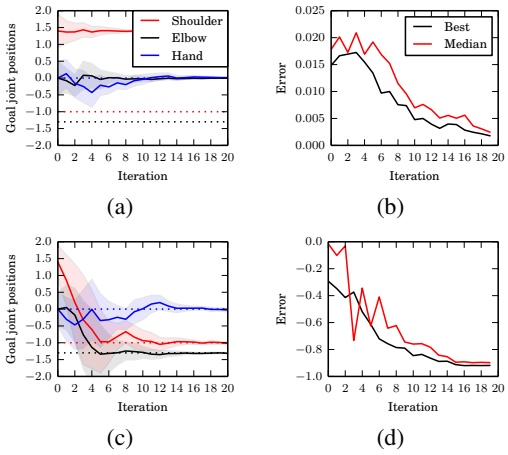

Fig. 6: Learning process for the first experiment, where a simulated schunk arm LWA 4P learns to move to a given pose demonstrated in simulation (see Fig. 3). We repeat this experiment two times with two different error metrics: the absolute difference (first row) and the negative Pearson correlation (second row). First column: means and standard deviations of goal joint positions for shoulder, elbow and hand joints. Ground truth is shown in dashed lines. (a): The arm learns not to move (initial joint positions equals learned joint positions). (c): The precise joint positions are recovered. Second column: decrease of the different error metrics for the two runs. In both runs, the error metric decreases, meaning that the robot has learned the movement with confidence. (b): Absolute error metric for the first run. (d): Negative Pearson Correlation error metric for the second run.

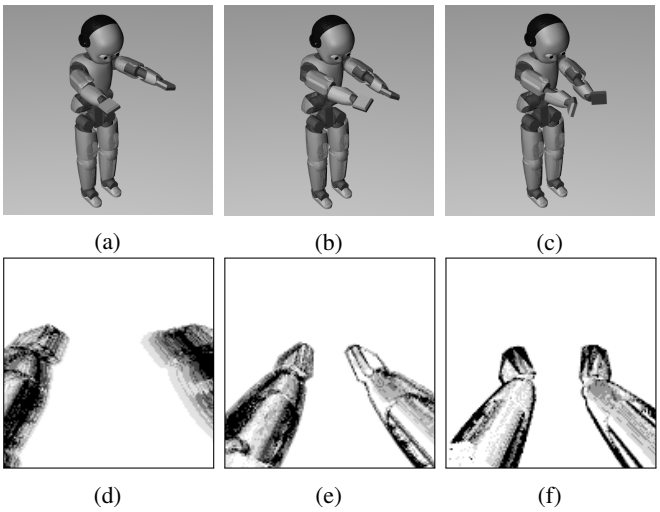

Fig. 7: Learned movement for the second experiment. the simulated iCub learned to move its arms closer together, as in the provided human demonstration (see Fig. 4). First row: third-person view at the beginning (a), during (b) and at the end (c) of the learned motion. The selected viewpoint is used for visualization purposes and is not used for training. Second row: aggregated DVS address events at the beginning (d), during (e) and at the end (f) of the learned motion.

The third experiment shows that multiple movements can be learned out of a single demonstration depending on visual cues. In this case, the liquid state machine managed to associate a visual cue to an arm movement, based on their time of appearance in the demonstration (see Fig. 5). Indeed, when the ball flashes on the left, the arm will be predicted on the left side, and when the ball flashes on the right, the arm will be predicted on the right side. This is confirmed by the successful optimization (Fig. 9) recovering visually similar arm movements depending on presented visual cues (Fig. 10). Conversely, if the arm moves to one side, the liquid state machine will predict the ball flashing on this side: our method does not differentiate in the visual input between what it can control and what it can not.

V. CONCLUSIONS

In this paper, we introduced a method for robots to reproduce movements visually similar to a demonstration, by minimizing event-based visual prediction error. The method has two phases. In the first phase, we train a spiking neural network to predict future visual input from a demonstration. This prediction model represents the memory of the demonstrated movement. In the second phase, the robot tries to find the movement most visually similar to the predictable one.

In this paper, visual information is represented with event streams instead of sequences of frames. Therefore, only changes in the scene (motions) are sensed, while redundant information are not processed. We propose to evaluate the visual similarity between event streams using the Pearson correlation. We show how this definition enables movements to be learned, unlike the definition proposed in [13]. Our approach is able to recover precise goal joint positions (Fig. 6c), can scale up to potentially many joints even when demonstrations are provided by humans (Fig. 8), and can learn different movements from a single demonstration depending on visual cues (Fig. 10).

For future work, one could use the prediction error as a reinforcement signal instead of a fitness function. Indeed, by using reinforcement learning methods instead of evolutionary strategy, movements could be learned online in closed-loop. In our work, prediction error was computed per timestep (Equation 3), but aggregated for a whole sequence for the fitness evaluation (Equation 4). With reinforcement learning algorithms, the temporal structure of the error could also be taken into account, theoretically speeding up learning. Recently, the similarity between CMA-ES and the state-of-the-art policy search method PI^2 [34] has been investigated [35]. In a reinforcement learning setup, our method could also control the robot continuously instead of learning a single movement. Spiking network implementations for reinforcement learning [36] and activation of motion primitives [37, 38] have already been proposed. This way, our method could be implemented in a purely event-driven fashion, relying solely on spiking networks.

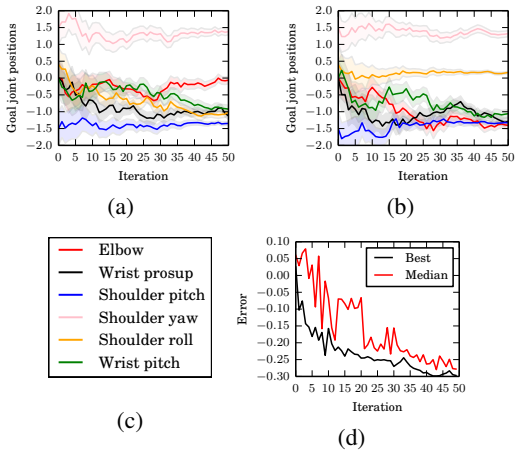

Fig. 8: Learning process for the second experiment, where a simulated iCub robot learns to move both arms to a given pose demonstrated by a human (see Fig. 4). (a, b): Means and standard deviations of goal joint positions for the six joints of the left (a) and right (b) arms, as labeled in (c). Learned joint goal positions are similar for left and right arm, except for the elbow and the shoulder roll, indicating a symmetry in the learned movement. (d): Decrease of the negative Pearson correlation over iterations.

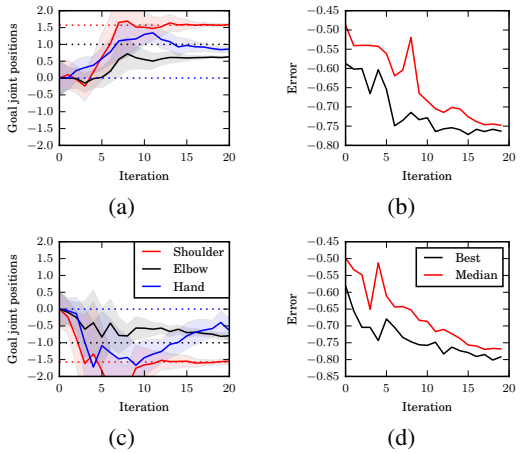

Fig. 9: Learning process for the third experiment, where a simulated schunk arm LWA 4P learns two movements depending on visual cues (see Fig 5). We repeat this experiment two times: the first time, the ball flashes on the left, the second time it flashes on the right. First column: means and standard deviations of goal joint positions for shoulder, elbow and hand joints. Ground truth is shown in dashed lines. (a): The arm learns to move left (see Fig. 10a). (c): The arm learns to move right. (see Fig. 10c) Second column: decrease of the different error metrics for the two runs. In both runs, the error metric decreases, meaning that the robot has learned the movement with confidence. (b, d): Decrease of the negative Pearson correlation over iterations.

Another interesting improvement would be to learn the visual prediction model differently than by imitation. For this purpose, one could feed to the liquid state machine not only visual input but also a copy of the motor command sent to the joints. This way, the robot could learn the perception of its own body. To remove the constraint that the demonstrations have to be provided in first-person view, one could learn a mapping between the demonstrator and the robot. In biology, such a mapping is believed to originate from mirror neurons [39].

ACKNOWLEDGMENT

This research has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement No. 720270 (Human Brain Project SGA1) and No. 785907 (Human Brain Project SGA2).

REFERENCES

[1] A. H. Marblestone *et al.*, "Towards an integration of deep learning and neuroscience," *bioRxiv*, vol. 10, no. September, pp. 1–61, 2016. arXiv: 1606.03813.

[2] J. L. Copete *et al.*, "Motor development facilitates the prediction of others' actions through sensorimotor predictive learning," in *Joint Int. Conf. on Development and Learning and Epigenetic Robotics*, IEEE, 2016, pp. 223–229.

[3] J. Hohwy, *The predictive mind*. Oxford University Press, 2013.

[4] M. V. Butz, "Towards a unified sub-symbolic computational theory of cognition," *Frontiers in Psychology*, vol. 7, no. 925, 2016.

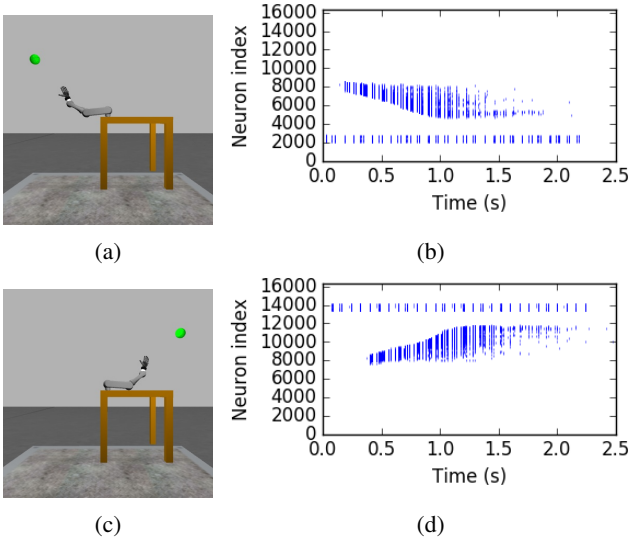

Fig. 10: Learned movements for the third experiment. (a, c): The movements learned by the arm depend on the visual cue and are visually similar to the demonstrated ones. (b, d): Input spiketrains of the learned movements.

[5] K. Friston, "The free-energy principle: A unified brain theory?" *Nature Reviews Neuroscience*, vol. 11, no. 2, pp. 127–138, 2010.

[6] M. V. Butz *et al.*, Eds., *Anticipatory Behavior in Adaptive Learning Systems: Foundations, Theories, and Systems (LNAI 2684)*. Springer Verlag, 2003.

[7] D. M. Wolpert *et al.*, "Motor prediction," *Current biology*, vol. 11, no. 18, R729–R732, 2001.

[8] S. Otte *et al.*, "Inferring adaptive goal-directed behavior within recurrent neural networks," *Int. Conf. on Artificial Neural Networks*, pp. 227–235, 2017.

[9] R. Der *et al.*, "Novel plasticity rule can explain the development of sensorimotor intelligence," *Proceedings of the National Academy of Sciences*, vol. 112, no. 45, E6224–E6232, 2015.

[10] D. M. Wolpert *et al.*, "Multiple paired forward and inverse models for motor control," *Neural Networks*, vol. 11, no. 7-8, pp. 1317–1329, 1998.

[11] J. R. Flanagan *et al.*, "Prediction precedes control in motor learning," *Current Biology*, vol. 13, no. 2, pp. 146–150, 2003.

[12] C. Finn *et al.*, "Deep Visual Foresight for Planning Robot Motion," 2016. arXiv: 1610.00696.

[13] J. Kaiser *et al.*, "Scaling up liquid state machines to predict over address events from dynamic vision sensors," *Bioinspiration & Biomimetics*, 2017.

[14] N. Hansen *et al.*, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.

[15] B. Roska *et al.*, "Rapid global shifts in natural scenes block spiking in specific ganglion cell types," *Nature neuroscience*, vol. 6, no. 6, p. 600, 2003.

[16] N. Kruger *et al.*, "Deep hierarchies in the primate visual cortex: What can we learn for computer vision?" *Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1847–1871, 2013.

[17] J. Kaiser *et al.*, "Learning movements by imitation from event-based visual prediction," in *2nd HBP Student Conference*, (Extended Abstract), 2018.

[18] N. Srivastava *et al.*, "Unsupervised learning of video representations using lstms," in *Int. Conf. on Machine Learning*, 2015, pp. 843–852.

[19] M. Mathieu *et al.*, "Deep multi-scale video prediction beyond mean square error," *CoRR*, vol. abs/1511.05440, 2015.

[20] N. Sedaghat, "Next-flow: Hybrid multi-tasking with next-frame prediction to boost optical-flow estimation in the wild," *CoRR*, vol. abs/1612.03777, 2016.

[21] H. Akolkar *et al.*, "What can neuromorphic event-driven precise timing add to spike-based pattern recognition?" *Neural Computation*, vol. 27, no. 3, pp. 561–593, 2015.

[22] W. Maass *et al.*, "A new approach towards vision suggested by biologically realistic neural microcircuit models," *Neural Computation*, vol. 2525, p. 282 293, 2002.

[23] H Burgsteiner *et al.*, "Movement prediction from real-world images using a liquid state machine," *Applied Intelligence*, vol. 26, no. 2, pp. 99–109, 2007.

[24] D. Pathak *et al.*, "Curiosity-driven exploration by self-supervised prediction," *CoRR*, vol. abs/1705.05363, 2017.

[25] J. Lee *et al.*, "Learning robot activities from first-person human videos using convolutional future regression," *CoRR*, vol. abs/1703.01040, 2017.

[26] A. W. Tow *et al.*, "What would you do? acting by learning to predict," *CoRR*, vol. abs/1703.02658, 2017.

[27] R. Der *et al.*, "Self-organized behavior generation for musculoskeletal robots," *Frontiers in Neurorobotics*, vol. 11, p. 8, 2017.

[28] W. Maass, "Liquid state machines: Motivation, theory, and applications," *Computability in Context: Computation and Logic in the Real World*, pp. 275–296, 2010.

[29] W Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural networks*, vol. 10, no. 9, pp. 1659–1671, 1997.

[30] E. Falotico *et al.*, "Connecting artificial brains to robots in a comprehensive simulation framework: The neurorobotics platform," *Frontiers in Neurorobotics*, vol. 11, p. 2, 2017.

[31] P. Lichtsteiner *et al.*, "A 128x128 120 db 15us latency asynchronous temporal contrast vision sensor," *journal of solid-state circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[32] E. Mueggler *et al.*, "Event-based , 6-dof pose tracking for high-speed maneuvers," in *Int. Conf. on Intelligent Robots and Systems*, IEEE, 2014.

[33] J. Kaiser *et al.*, "Towards a framework for end-to-end control of a simulated vehicle with spiking neural networks," in *Int. Conf. on Simulation, Modeling, and Programming for Autonomous Robots*, IEEE, 2016, pp. 127–134.

[34] E. Theodorou *et al.*, "A generalized path integral control approach to reinforcement learning," *The Journal of Machine Learning Research*, vol. 11, pp. 3137–3181, 2010.

[35] F. Stulp *et al.*, "Robot skill learning: From reinforcement learning to evolution strategies," *Paladyn, Journal of Behavioral Robotics*, vol. 4, no. 1, pp. 49–61, 2013.

[36] D. Kappel *et al.*, "Reward-based stochastic self-configuration of neural circuits," 2017. eprint: arXiv:1011.1669v3.

[37] J. C. V. Tieck *et al.*, "Multi-modal motion activation for robot control using spiking neurons," in *Int. Conf. on Biomedical Robotics and Biomechatronics*, IEEE, 2018, Submitted.

[38] J. C. Vasquez Tieck *et al.*, "Towards Grasping with Spiking Neural Networks for an Anthropomorphic Robot Hand," in *Int. Conf. on Artificial Neural Networks*, 2017.

[39] G. d. Pellegrino *et al.*, "Understanding motor events: A neurophysiological study," *Experimental brain research*, vol. 91, no. 1, pp. 176–180, 1992.