2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Probabilistic Spiking Neural Network for Robotic Tactile Continual Learning

Senlin Fang[1,2], Yiwen Liu[2], Chengliang Liu[2], Jingnan Wang[2], Yuanzhe Su[2], Yupo Zhang[2], Hoiio Kong[1], Zhengkun Yi[2] and Xinyu Wu[2]

***Abstract*— The sense of touch is essential for robots to perform various daily tasks. Artificial Neural Networks have shown significant promise in advancing robotic tactile learning. However, due to the changing of tactile data distribution as robots encounter new tasks, ANN-based robotic tactile learning suffers from catastrophic forgetting. To solve this problem, we introduce a novel continual learning (CL) framework called the Probabilistic Spiking Neural Network with Variational Continual Learning (PSNN-VCL). In this framework, PSNN introduces uncertainty during spike emission and can apply fast Variational Inference by optimizing the uncertainty through backpropagation, which significantly reduces the required model parameters for VCL. We establish a robotic tactile CL benchmark using publicly available datasets to evaluate our method. Experimental results demonstrated that, compared to other CL methods, PSNN-VCL not only achieves superior performance in terms of widely used CL metrics but also achieves at least a 50% reduction in model parameters on the robotic tactile CL benchmark.**

I. INTRODUCTION

The sense of touch is essential for robots to perform various daily tasks, such as object recognition [1], hardness classification [2], and slip detection [3]. Artificial neural networks (ANNs) have shown significant promise in advancing robotic tactile perception, allowing agents to achieve robust and accurate tactile perception. In practice, a robot is expected not only to perform well on current tasks but also to maintain performance achieved on previous tasks. Traditionally, to ensure the high performance of the ANN model across all tasks, the model is trained to simultaneously learn all tasks through multi-task learning [4]. However, this approach is infeasible in the field of robotics due to the impracticality of a robot gathering data from all tasks at the same time. It seems more natural and convenient for a robot to continuously learn than to try to learn all tasks at the same time.

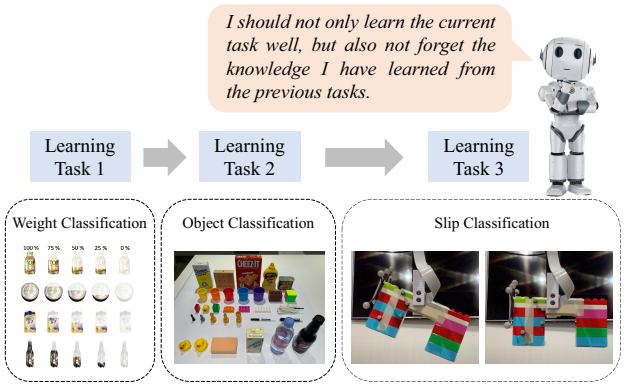

Fig. 1. Schematic of Robotic Tactile Continual Learning. In this paper, we consider one of the continual learning scenarios, which is task-incremental learning.

In real-life scenarios, the tactile information (hardness, shape, texture, etc.) presented to a robot evolves with the emergence of new tasks. An ideal embodied agent should perform well in a current task while maintaining the performance of previous tasks. However, ANN models suffer from catastrophic forgetting, that is, when they are trained on a new task, they quickly forget what they have learned from previous tasks [5]. As a result, robots are unable to consistently learn and accumulate knowledge when utilizing ANN models. To tackle this issue, Continual learning (CL) strategies have been proposed to make the ANN model perform well on new tasks while maintaining the performance achieved in previous tasks to the greatest extent. CL allows robots to consistently accumulate tactile knowledge from various tasks, thereby potentially enabling robots to learn continuously like humans. In this paper, we consider one of the CL scenarios, task-incremental learning in the robotic tactile field, see Fig. 1. In the field of robotics, researchers typically employ two CL methods to solve the catastrophic forgetting problem: regularization-based methods [6]–[8], and memory replay methods [9]–[12]. Regularization-based methods introduce additional reg-

The work described in this paper is partially supported by the National Natural Science Foundation of China (No. 62125307 and No. U22A2056), partially supported by the Science and Technology Innovation Commission of Shenzhen (No. JSGGZD20220822095401004 and No. JCYJ20220818101205011). (Corresponding author: Zhengkun Yi)

[1]Senlin Fang and Hoiio Kong are with City University of Macau, Macau, 999078, China. E-mails: {d22092100360, hikong}@city.mo

[2]Senlin Fang, Yiwen Liu, Chengliang Liu, Jingnan Wang, Yuanzhe Su, Yupo Zhang, Zhengkun Yi and Xinyu Wu are with the the Department of Intelligent Systems and Robot Learning Lab ISRL group, SIAT Branch, Institute of Artificial Intelligence and Robotics for Society, Shenzhen Institute of Advanced Technology, Shenzhen, 518055. E-mails: {sl.fang1, yw.liu, cl.liu1, jn.wang, yz.su, yp.zhang1, zk.yi, xy.wu}@siat.ac.cn

ularization terms to penalize significant changes in important parameters while learning new tasks [13]. For example, Knoedler et al. [6] applied Elastic Weight Consolidation (EWC) to enable autonomous mobile robots to continuously predict pedestrian motion. Pique et al. [7] employed EWC to enable soft robots to continuously adapt to different external loads. Zheng et al. [8] introduced a distillation loss constraint method to achieve visual-tactile cross-modal material continual perception. Memory replay methods alleviate knowledge forgetting by replaying past task data [14]. For instance, Chruamani et al. [9] introduced a Growing Dual-Memory (GDM) architecture that enables robots to incrementally recognize each individual's expressions. Soh et al. [10] designed a generative learning network to generate pseudo-samples, facilitating incremental tactile recognition tasks for robots. Chen et al. [11] employed a generative model for experience replay, which enables robots to continuously learn in various physical environments. Xiong et al. [12] proposed a primitive generation approach for tasks like robot door opening and peg insertion in simulated environments.

However, regularization methods often exhibit lower performance compared to other CL methods [15], [16]. Memory replay methods demand high storage and computational capabilities, as they require storing data from previous tasks. To solve this problem, Nguyen et al. [17] proposed a Variational Continual Learning (VCL) method, which uses Variational Inference (VI) [18] to update the posterior distribution from previous tasks. VCL stands out by not necessitating the replay of past samples and has demonstrated superior performance compared to other CL methods on various computer vision benchmarks [17]. Regrettably, VCL, due to its reliance on VI, leads to a doubling of the model's parameters. In this paper, we introduce a novel CL framework, namely the Probabilistic Spiking Neural Network with Variational Continual Learning (PSNN-VCL) to solve this problem. The main contributions of this work are as follows.

- A novel CL framework, namely PSNN-VCL tailored for robotic tactile CL scenario is proposed. The framework can not only make agents work well for the new tasks but also preserve performance on the previous task without the need to replay the previous task samples.
- A novel PSNN model is developed, which introduces uncertainty during spike emission. This innovation allows PSNN to not only learn the uncertainty inherent in time-series data but also apply fast VI by optimizing this uncertainty through backpropagation, which significantly reduces the required model parameters for VCL.
- We establish a robotic tactile CL benchmark using publicly available datasets. Experimental results demonstrated that compared with other CL baselines, our method not only achieves the best performance in terms of widely used CL metrics but also reduces model parameters by at least 50% on the robotic tactile CL benchmark.

II. RELATED WORK

A. Spiking Neural Network in Robotic Tactile Field

Spiking Neural Network (SNN) is a type of ANN that models the behavior of biological neurons more closely compared to traditional ANNs. In SNN, information is transmitted between neurons through spike trains (binary time series data). When the membrane potential exceeds a threshold, the neuron in SNN will generate a spike (0 or 1), also called firing.

There is a lot of research applying SNN in the robotic tactile field. For example, Friedl et al. [19] used adaptive leaky-integrate-and-fire (LIF) neurons to extract high-dimension features. The features are used to classify 18 metal surface textures, achieving a 65.6% classification accuracy. Yi et al. [19] applied the Izhikevich model to generate spike trains. The features extracted from the generated spike are used to classify 8 different textures, resulting in an accuracy of approximately 77.6%. Taunyazov et al. [20] used the SNN to classify 20 material textures, resulting in faster inferences as compared to state-of-the-art (SOTA) learning approaches. Taunyazov et al. [29] recorded event-based tactile and visual datasets using a robot system. Subsequently, they employed an SNN model to fuse and train on data from both modalities. Their experiments demonstrated excellent performance in daily-life object classification, container weight classification, and slip detection tasks. Additionally, Gu et al. [21] leveraged the event-based datasets to learn the spatial feature among 39 tactile taxels data using spike graph neural networks. Kang [22] further proposed the Time Spike Response Model, which can capture spatio-temporal patterns of the tactile data. To the best of our knowledge, we are the first to apply SNN in the robotic tactile CL scenario.

III. PROPOSED METHOD

A. Problem Setting

Consider a collection of datasets $\{D^{(1)}, \ldots, D^{(N)}\}$ for N different tactile classification tasks. Each dataset $D^{(n)}$ is represented as $\{(x_k^{(n)}, y_k^{(n)})_{k=1}^{K_n}\}$, where $x_k^{(n)}$ represents the kth sample collected in the task n, $y_k^{(n)}$ denotes the corresponding labels, and K_n is the sample number of the task n. The data distribution changes as the task changes. The agent encounters tasks sequentially, without revisiting data in previous tasks. The objective is to ensure the agent continually adapt to the

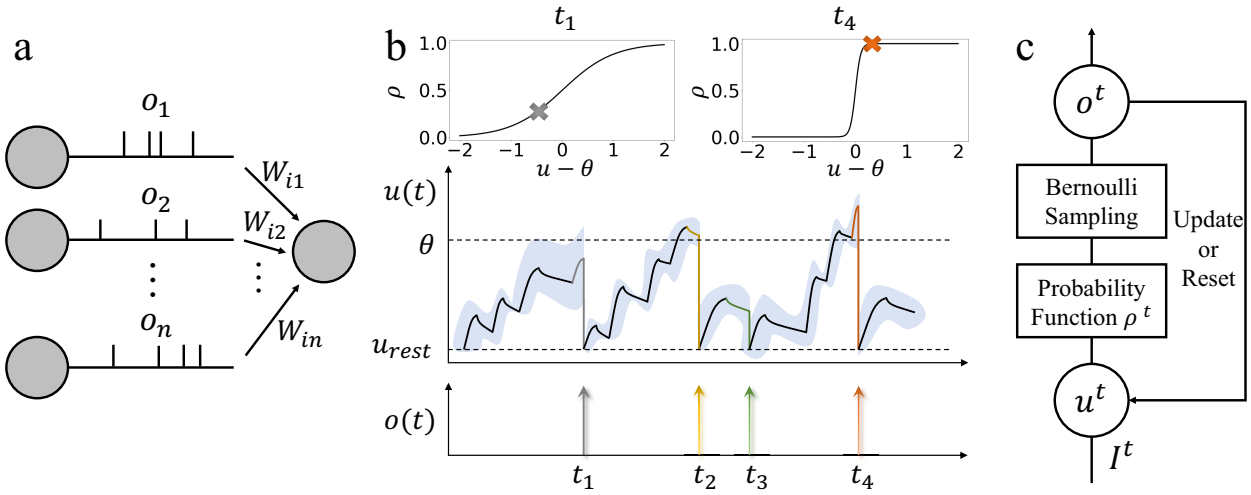

Fig. 2. Architecture of Probabilistic Spiking Neural Network. (a) Presynaptic input $I(t)$ is the weighted sum of spikes from the previous layer; (b) PSNN can generate spikes when u has not yet reached threshold u_{th} and remain silent even when u has exceeded u_{th}. The shaded blue area represents the uncertainty learned by PSNN, which is inversely proportional to the parameter controlling the steepness of the probability function curve. A larger shaded blue area indicates higher uncertainty, resulting in a smoother curve; (c) The computational flowchart illustrating PSNN firing.

data encountered in the new task while maintaining its performance on previous tasks.

B. PSNN

Although there are several notable SNN models, including the Hodgkin-Huxley model proposed by Hodgkin and Huxley [23], Izhikevich model by Izhikevich [24], the Tempotron model by Robert and Haim [25], these models either have high computational costs, or they exhibit performance gaps compared to traditional ANNs. The PSNN is bulit upon the iterative LIF model, which is proposed by Wu et al. [26] can achieve training acceleration by orders of magnitude for deep SNNs. Zhang et al. [27] and Zheng et al. [28] leveraged this iterative LIF to achieve high classification performance on Computer Vision datasets.

The architecture of PSNN is shown in Fig. 2. Specifically, the iterative LIF model in PSNN employs the Euler method for integration as follows.

$$u_i(t) = \tau u_i(t-1)(1-o_i(t-1)) + I_i(t), \quad (1)$$

where τ is a fixed decay constant. For layer i, $u_i(t)$ represents the membrane potential at time t, and $o_i(t-1)$ is the binary input spike at time $t-1$. See Fig. 2a, $I_i(t)$ denotes the presynaptic input at time t, which is the weighted sum of input spikes from the previous layer:

$$I_i(t) = \sum_j w_{i,j} o_{(i-1,j)}(t). \quad (2)$$

Here, the suberscript j represents the index of the presynaptic neuron, and $w_{i,j}$ signifies the network weight. In traditional iterative LIF model, neuron firing in layer i when $u_i(t)$ exceeds a constant membrane potential threshold u_{th}:

$$o_i(t) = H(u_i(t) - u_{th}), \quad (3)$$

where $u_i(t)$ represents the membrane potential of the neuron in layer i at time t. Nevertheless, this approach comes with several drawbacks. Firstly, due to the non-differentiability of Eq. 3, the surrogate function for backpropagation is always required. Secondly, this deterministic firing pattern is incapable of capturing uncertainty.

The PSNN addresses this issue by introducing an escape noise mechanism to the iterative LIF model. In the escape noise mechanism, neurons do not fire based on whether they exceed a threshold; instead, neurons may fire even when they have not exceeded the threshold, and they may remain silent even when they have exceeded it. For instance, see Fig. 2b, the PSNN can generate a spike when $u(t_1)$ has not yet reached threshold u_{th}, and it can even spike when $u(t_3)$ is significantly below u_{th}. The neuron can remain quiescent, even if $u(t_2)$ and $u(t_4)$ have crossed the u_{th}.

In detail, the computational flowchart illustrating PSNN firing is presented in Fig. 2c. Initially, $u_i(t)$ is transformed into the escape probability $\rho(t)$ using the Logistic function as follows.

$$\rho(t) = \frac{1}{1 + e^{-(u_i(t) - V_{th})/\sigma_i(t)}}. \quad (4)$$

Here $\sigma_i(t)$ is a factor used to control the steepness of the Logistic function curve, and it is also inversely proportional to the degree of uncertainty. As shown in Fig. 2b, the shaded blue area represents the degree of

uncertainty learned by PSNN. A larger area indicates higher uncertainty at the current time step, while a smaller area implies lower uncertainty. The $\sigma_i(t_1)$ is larger than the $\sigma_i(t_2)$, resulting in a flatter curve, which means the PSNN has higher uncertainty at t_1 than t_2. Finally, Bernoulli sampling is performed with probability $\rho(t)$ to obtain $o_i(t)$:

$$o_i(t) \sim \text{Bernoulli}\left(\rho(t)\right). \tag{5}$$

Algorithm 1 PSNN-VCL

Require: Training data for current task $\mathcal{D}^{(n)} \leftarrow (x^{(n)}, y^{(n)})$, posterior distribution $q(\theta|\mathcal{D}^{(1:n-1)})$ learned from previous tasks

1: Init: $p(\theta) \sim \mathcal{N}(0,1)$, $o_0^{(n)} \leftarrow x^{(n)}$
2: **for** $n = 1 \dots N$ **do**
3: **if** $n = 1$ **then**
4: Prior $\leftarrow p(\theta)$
5: **else**
6: Prior $\leftarrow q(\theta|\mathcal{D}^{(1:n-1)})$
7: **end if**
8: **for** $i = 1 \dots I$ **do**
9: $\sigma_i(t) \leftarrow \sqrt{\sum_j \left(w\_\sigma_{(i,j)}\right)^2 \left(o_{(i-1,j)}^{(n)}(t)\right)^2}$
10: $I_i(t) \leftarrow \sum_j w_{(i,j)} - \mu_{(i,j)} o_{(i-1,j)}^{(n)}(t)$
11: $u_i(t) \leftarrow \tau u_i(t-1)\left(1 - o_i^{(n)}(t-1)\right) + I_i(t)$
12: $k_i(t) \leftarrow \frac{1}{\sigma_i(t)}$
13: $\rho_i(t) \leftarrow \text{logistic}\left(u_i(t), k_i(t)\right)$
14: $o_i^n(t) \leftarrow \text{Ber}(\rho_i(t))$
15: **end for**
16: $u_I(T) \leftarrow \sum_{t=1}^{T} \lambda o_i^{(n)}(t)$
17: $q(\mathcal{D}_n|\theta) \leftarrow \text{Softmax}\left(u_I(T)\right))$
18: $q(\theta|\mathcal{D}^{(1:n)}) \propto \text{Prior} \cdot \text{q}(\mathcal{D}^{(1:n-1)}|\theta)$
19: **end for**

C. PSNN-VCL

To address the problem of robotic tactile CL in Sec. III-A, we introduced the PSNN-VCL framework, which combines PSNN with the VCL method [17]. VCL performs task-specific updates to the posterior distribution using the Bayes rule:

$$\text{q}(\theta|\mathcal{D}^{(1:n)}) \propto \underbrace{\text{p}(\mathcal{D}^{(n)}|\theta)}_{\substack{\text{likelihood learned} \\ \text{by current task}}} \underbrace{\text{q}(\theta|\mathcal{D}^{(1:n-1)})}_{\substack{\text{posterior from} \\ \text{previous task}}}. \tag{6}$$

Here the prior distribution for the current task is the posterior distribution learned from the previous tasks. By updating the prior distribution using the likelihood distribution after each task training, we can obtain the posterior distribution for all observed tasks. However, the posterior distribution is intractable, and VCL employs a VI approach for approximation [17].

The PSNN-VCL method is outlined in Alg. 1. Specifically, for the initial task D_1, we initialize the prior distribution $p(\theta)$ as a normal distribution, as shown in line 4 of Alg. 1. For non-initial tasks D_n, their prior distribution is the posterior distribution of the previous task, as shown in line 6 of Alg. 1. The computation steps for obtaining the PSNN-VCL likelihood distribution are as follows.

For each layer of PSNN-VCL, at time step t, two main parameters, $u(t)$ and $k(t)$, are learned. The logistic function from Eq. 4 is used to calculate the firing probability at the current time step. Here $u(t)$ controls the firing probability: larger $u(t)$ result in higher firing probabilities, while smaller $u(t)$ result in lower probabilities. $k(t)$ controls the uncertainty of firing probabilities: larger $k(t)$ yield a flatter logistic function, making $\rho(t)$ closer to 50% (as shown $\rho(t_1)$ in Figure 2b), while smaller $k(t)$ values result in a steeper logistic function, resembling a step function, making $\rho(t)$ closer to 0% or 100% (as shown $\rho(t_4)$ in Figure 2b).

Through Bernoulli sampling of the spike trains and layer-wise propagation of the hidden layer distribution, we obtain:

$$q\left(o_{n+1}(t) \mid o_1(\text{t}), \theta(\text{t})\right) = \prod_1^n \text{q}\left(o_{n+1}(\text{t}) \mid o_n(t), \theta(\text{t})\right), \tag{7}$$

where $\theta(t)$ is the variational paramters $\{u(t), k(t)\}$. Following the steps in lines 16 and 17 of Alg. 1, we obtain the likelihood distribution of the neural network for the current task. Finally, similar to [17], we update posteriors by applying VI (lines 18 of Alg. 1), which is equal to minimize the following objective loss function:

$$\begin{aligned}\theta^* = \arg\min_\theta \big\{ &-E_{\theta\sim q(\theta)}\left[\log p\left(\mathcal{D}^{(n)} \mid \theta\right)\right] \\ &+KL\left(q(\theta)||q\left(\theta \mid \mathcal{D}^{(1:n-1)}\right)\right)\big\},\end{aligned} \tag{8}$$

where the first term is the cross-entropy loss, and the second term is the KL divergence between the parameter distribution of the current task with the posterior from the previous task. PSNN can learn the optimal θ^* through backpropagation.

IV. EXPERIMENTS

The proposed PSNN-VCL is evaluated on three publicly available datasets. First, the adopted datasets, baseline methods, evaluation metrics, and implementation details are introduced. Then, the experiment results and their analysis are given.

TABLE I

A SUMMARY OF TRAINING AND TESTING DATA DIMENSIONS FOR EACH TASK IN THE ROBOTIC TACTILE CL EXPERIMENT. THE THREE TASKS INVOLVE DIFFERENT NUMBERS OF CLASSES AND SAMPLES FOR CLASSIFICATION, WHILE THE SPATIAL AND TEMPORAL DIMENSIONS OF THE INPUT DATA ARE THE SAME.

	Container	Object	Slip
Class number	20	36	2
Training sample number	640	720	80
Test sample number	160	180	20
Input feature size	78	78	78
Number of timesteps	150	150	150

A. Datasets

For the experiments on robotic tactile CL, we utilized three publicly available datasets collected by NeuTouch [29]. We refer to these datasets as Object, Container, and Slip. The data collection for all three datasets employed the Franka Emika Panda robotic arm equipped with a Robotiq 2F-140 gripper. Two NeuTouch event-based tactile sensors were mounted on the gripper, with each sensor comprising 39 taxels. Sensor signals were transformed into spike trains through the ACES decoder. The specific details of the three datasets are as follows.

- Container: This dataset was used for tactile water level classification in containers. It included four different containers, each filled with water to levels of {0%, 25%, 50%, 75%, 100%}. Thus, there were 20 different classes. To collect this dataset, the robotic gripper initially grasped the containers and then lifted them to a height of 5 cm above the table. Each class was repeatedly collected 40 times, yielding a total of 800 samples.
- Object: This dataset was employed for tactile object classification, consisting of 36 daily life objects. To collect this dataset, the robotic gripper first grasped these objects and then lifted them to a height of 20 cm above the table surface before placing them back on the table. Each object was grasped 15 times, resulting in a total of 900 samples.
- Slip: This dataset was designed for tactile slip detection and consisted of two classes (slip or no-slip). One class involved grasping Lego bricks that were symmetrically weighted, while the other class had a hidden 10g mass attached to either the left or right leg of the Lego brick, rendering the two classes indistinguishable visually. The robotic gripper separately grasped and lifted both types of objects to a height of 10 centimeters above the table. Each class was collected 50 times, resulting in a total of 100 samples.

In the three datasets, we selected the first 150 time-step data for each dataset. Data were randomly split into training sets (80%) and testing sets (20%). The summary of the data used to train and test is shown in Tab. I. In this tactile CL setting, we assume that the training data of three different tasks arrive in sequence as Container $\rightarrow$ Object $\rightarrow$ Slip. The training samples in previous tasks are not available when learning new tasks.

B. Baselines and Evaluation Metric

We compare our PSNN-VCL with the ANN-based CL baselines including ANN-EWC, ANN-SI, ANN-LwF, and ANN-VCL, each applying EWC, SI, LwF, and VCL methods, respectively. The lower bounds ANN-NAVIE and PSNN-NAVIE are included, representing ANN and PSNN models without using any continual learning strategies. Additionally, the upper bounds ANN-JT and PSNN are also included, representing ANN and PSNN models trained simultaneously on data from all tasks in a mult-tasks learning manner.

Similar to [15], ACC, BWT, and #Params are used as evaluation metrics to compare the CL performance in this experiment. ACC is the average test classification accuracy on all observed tasks, and BWT is backward transfer, which indicates how much learning the current task has influenced the performance of previous tasks. The higher the BWT means the better the anti-forgetting performance. #Params denotes the parameter number of a model, which measures the space complexity of the model. Formally, ACC and BWT are as follows.

$$\mathrm{ACC} = \frac{1}{N}\sum_{n=1}^{N} R_{n,N}, \quad \mathrm{BWT} = \frac{1}{N}\sum_{n=1}^{N}(R_{n,N} - R_{n,n}) \tag{9}$$

C. Implementation Details

To ensure a fair comparison, PSNN-VCL and all baseline models share the same network hyperparameters, with the exception that the PSNN naturally learns temporal features through fully connected layers, obviating the need for LSTM structures typically required by ANN models for temporal feature learning. In particular, the network structure is $156 \rightarrow 128 \rightarrow 128 \rightarrow c$, where c is the category number of the specific task. The four layers of PSNN are all fully connected layers. Except for the third layer of ANN, which is LSTM with 128 hidden sizes, all layers of ANN are fully connected layers.

In training time, the batch size is 20, the learning rate is 0.001, the epoch is 50, and the Adam optimizer is used. To verify the effectiveness of our proposed PSNN. Each experiment was repeated 10 times, and we reported the average ACC and variance.

D. Results

Fig. 3 shows the comparison of the ACC between our proposed PSNN-VCL and the baselines on the

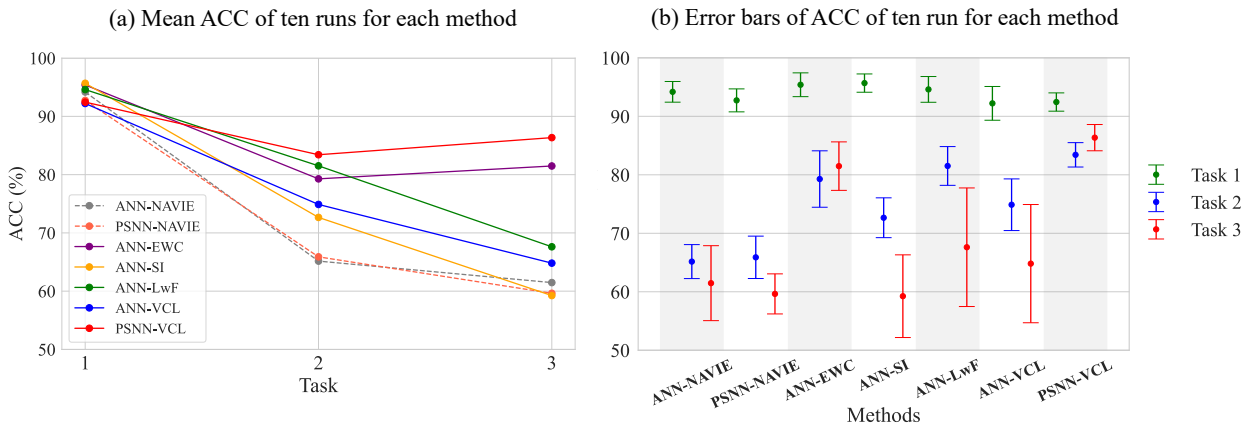

Fig. 3. Comparison of the ACC between our proposed PSNN-VCL and the baselines on the tactile CL benchmark. (a) presents the mean ACC across ten runs for each method. (b) displays the error bars representing the variability in ACC across these ten runs.

tactile CL benchmark. From Fig. 3a, we can observe that our proposed method, PSNN-VCL, achieves the highest accuracy (ACC) compared to other CL methods. Notably, the performance of the ANN-VCL method is poor. This suggests that achieving SOTA results on Computer Vision CL datasets may not necessarily translate to suitability for robotic tactile continual learning tasks when using the VCL approach. Conversely, our PSNN-VCL, employing the VCL approach, demonstrates better adaptability to robotic tactile continual learning tasks. Furthermore, as depicted in Fig. 3b, we observe that, compared to other methods, PSNN-VCL exhibits lower variance in ACC over ten experiments, indicating higher robustness in this experiment.

Tab. II provides all the metrics between our proposed PSNN-VCL and the baselines on the tactile CL benchmark. Notably, PSNN-VCL achieves the highest BWT of -0.13, signifying minimal knowledge forgetting when learning new tasks. Additionally, compared to ANN-based counterparts, PSNN-based methods, including PSNN-NAVIE, PSNN-VCL, and PSNN-JT, exhibit fewer model parameters. Specifically, PSNN-VCL's model parameters are only a quarter of ANN-VCL and half of other CL baselines (ANN-EWC, ANN-SI, and ANN-LwF). In summary, PSNN-VCL achieves superior performance on the tactile CL benchmark in terms of widely used CL metrics with lower model complexity, making it the most efficient method among all CL baselines.

TABLE II

COMPARISON OF THE METRICS BETWEEN OUR PROPOSED PSNN-VCL AND THE BASELINES ON THE TACTILE CL BENCHMARK (MEAN ± SEM). BECAUSE THE ANN-JT AND PSNN-JT USE THE MULTI-TASKS LEARNING STRATEGY, THE ACC AND BWT ARE NOT APPLICABLE (N/A) TO THESE TWO METHODS. THE BEST PERFORMANCE IS MARKED IN BOLD.

	Method	#Params	ACC(%)	BWT
Lower Bounds	ANN-NAVIE	0.152M	61.47±6.75	-0.52±0.07
	PSNN-NAVIE (ours)	0.036M	59.64±3.59	-0.55±0.05
CL Methods	ANN-EWC	0.152M	81.48±4.35	-0.18±0.05
	ANN-SI	0.152M	59.24±7.41	-0.51±0.09
	ANN-LwF	0.152M	67.62±10.63	-0.29±0.08
	ANN-VCL	0.304M	65.89±5.68	-0.40±0.05
	PSNN-VCL (ours)	**0.073M**	**86.35±2.35**	**-0.13±0.04**
Upper Bounds	ANN-JT	0.152M	95.04+2.09	N/A
	PSNN-JT (ours)	0.036M	95.92+0.85	N/A

V. CONCLUSIONS

In this paper, we introduce a novel PSNN-VCL method for robotic tactile CL. Specifically, we present a PSNN model capable of capturing the temporal uncertainty inherent in tactile data. The PSNN model is particularly well-suited for VI when combined with the VCL method. To evaluate the effectiveness of our approach, we establish a benchmark for robotic tactile CL and conduct experiments to validate our proposed method. The experimental results demonstrate that, in comparison to other CL methods, our approach not only achieves superior CL performance but also reduces the number of model parameters by a minimum of 50%. These findings offer an innovative and efficient solution to addressing continual learning challenges in the field of robotic tactile perception.

REFERENCES

[1] N. Jamali, P. Kormushev, A. C. Vinas, M. Carreras, and D. G. Caldwell, "Underwater robot-object contact perception using machine learning on force/torque sensor feedback," in *International Conference on Robotics and Automation*, 2015, pp. 3915–3920.

[2] S. Fang, Z. Yi, T. Mi, Z. Zhou, C. Ye, W. Shang, T. Xu, and X. Wu, "Tactonet: Tactile ordinal network based on unimodal probability for object hardness classification," *IEEE Transactions on Automation Science and Engineering*, 2022.

[3] J. Li, S. Dong, and E. Adelson, "Slip detection with combined tactile and visual information," in *International Conference on Robotics and Automation*, 2018, pp. 7772–7777.

[4] L. Pinto and A. Gupta, "Learning to push by grasping: Using multiple tasks for effective learning," in *International Conference on Robotics and Automation*, 2017, pp. 2161–2168.

[5] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *In Psychology of learning and motivation*, pp. 109–165, 2008.

[6] L. Knoedler, C. Salmi, H. Zhu, B. Brito, and J. Alonso-Mora, "Improving pedestrian prediction models with self-supervised continual learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4781–4788, 2022.

[7] F. Piqué, H. T. Kalidindi, L. Fruzzetti, C. Laschi, A. Menciassi, and E. Falotico, "Controlling soft robotic arms using continual learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5469–5476, 2022.

[8] W. Zheng, H. Liu, and F. Sun, "Lifelong visual-tactile cross-modal learning for robotic material perception," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 3, pp. 1192–1203, 2020.

[9] N. Churamani, M. Axelsson, A. Caldır, and H. Gunes, "Continual learning for affective robotics: A proof of concept for wellbeing," in *International Conference on Affective Computing and Intelligent Interaction Workshops and Demos*, 2022, pp. 1–8.

[10] H. Soh and Y. Demiris, "Incrementally learning objects by touch: Online discriminative and generative models for tactile-based recognition," *IEEE transactions on haptics*, vol. 7, no. 4, pp. 512–525, 2014.

[11] G. Chen, W. Zhang, H. Lu, S. Gao, Y. Wang, M. Long, and X. Yang, "Continual predictive learning from videos," in *International Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 728–10 737.

[12] F. Xiong, Z. Liu, K. Huang, X. Yang, and A. Hussain, "Primitives generation policy learning without catastrophic forgetting for robotic manipulation," in *International Conference on Data Mining Workshops*. IEEE, 2019, pp. 890–897.

[13] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *In Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[14] G. M. Van de Ven, H. T. Siegelmann, and A. S. Tolias, "Brain-inspired replay for continual learning with artificial neural networks," *Nature communications*, vol. 11, no. 1, p. 4069, 2020.

[15] S. Ebrahimi, M. Elhoseiny, T. Darrell, and M. Rohrbach, "Uncertainty-guided continual learning with bayesian neural networks," in *International Conference on Learning Representations*, 2020.

[16] M. Perkonigg, J. Hofmanninger, C. J. Herold, J. A. Brink, O. Pianykh, H. Prosch, and G. Langs, "Dynamic memory to alleviate catastrophic forgetting in continual learning with medical imaging," *Nature communications*, vol. 12, no. 1, p. 5678, 2021.

[17] C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner, "Variational continual learning," in *International Conference on Learning Representations*, 2018.

[18] Z. Ghahramani and H. Attias, "Online variational bayesian learning," in *Slides from talk presented at NIPS workshop on Online Learning*, 2000.

[19] K. E. Friedl, A. R. Voelker, A. Peer, and C. Eliasmith, "Human-inspired neurorobotic system for classifying surface textures by touch," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 516–523, 2016.

[20] T. Taunyazov, Y. Chua, R. Gao, H. Soh, and Y. Wu, "Fast texture classification using tactile neural coding and spiking neural network," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9890–9895.

[21] F. Gu, W. Sng, T. Taunyazov, and H. Soh, "Tactilesgnet: A spiking graph neural network for event-based tactile object recognition," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 9876–9882.

[22] P. Kang, S. Banerjee, H. Chopp, A. Katsaggelos, and O. Cossairt, "Event-driven tactile learning with location spiking neurons," in *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022, pp. 1–9.

[23] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of physiology*, vol. 117, no. 4, p. 500, 1952.

[24] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Transactions on neural networks*, vol. 14, no. 6, pp. 1569–1572, 2003.

[25] R. Gütig and H. Sompolinsky, "The tempotron: a neuron that learns spike timing–based decisions," *Nature neuroscience*, vol. 9, no. 3, pp. 420–428, 2006.

[26] Y. Wu, L. Deng, G. Li, J. Zhu, Y. Xie, and L. Shi, "Direct training for spiking neural networks: Faster, larger, better," in *Proceedings of the AAAI conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 1311–1318.

[27] W. Zhang and P. Li, "Temporal spike sequence learning via backpropagation for deep spiking neural networks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 022–12 033, 2020.

[28] H. Zheng, Y. Wu, L. Deng, Y. Hu, and G. Li, "Going deeper with directly-trained larger spiking neural networks," in *Proceedings of the AAAI conference on Artificial Intelligence*, vol. 35, no. 12, 2021, pp. 11 062–11 070.

[29] T. Taunyazov, W. Sng, H. H. See, B. Lim, J. Kuan, A. F. Ansari, B. C. Tee, and H. Soh, "Event-driven visual-tactile sensing and learning for robots," in *Proceedings of Robotics: Science and Systems*, 2020.