Split Additive Manufacturing for Printed Neuromorphic Circuits

Haibin Zhao[1], Michael Hefenbrock[2], Michael Beigl[1], and Mehdi B. Tahoori[1]
[1]*Karlsruhe Institute of Technology,* [2]*RevoAI*
[1]{haibin.zhao, michael.beigl, mehdi.tahoori}@kit.edu, [2]michael.hefenbrock@revoai.de

***Abstract*—Printed and flexible electronics promises smart devices for application domains, such as smart fast moving consumer goods and medical wearables, which are generally untouchable by conventional rigid silicon technologies. This is due to their remarkable properties such as flexibility, non-toxic materials, and having low-cost per area. Combined with neuromorphic computing, printed neuromorphic circuits pose an attractive solution for these application domains. Particularly, the additive printing technologies can reduce large amount of fabrication complexities and costs. On the one hand, high-throughput additive printing processes, such as roll-to-roll printing, can reduce the per-device fabrication time and cost. On the other hand, jet-printing can provide point-of-use customization at the expense of lower fabrication throughput. In this work, we propose a machine learning based design framework, that respects the objective and physical constraints of split additive manufacturing for printed neuromorphic circuits. With the proposed framework, multiple printed neural networks are trained jointly with the aim to sensibly combine multiple fabrication techniques (e.g., roll-to-roll and jet-printing). This should lead to a cost-effective fabrication of multiple different printed neuromorphic circuits and achieve high fabrication throughput, lower cost, and point-of-use customization.**

I. Introduction

As emerging domains, such as wearable computing [1], near-sensor computing [2], and edge AI [3] are unceasingly explored, the flexibility, non-toxicity, and low cost are desperately in demand. In this regard, printed electronics (PE) becomes one of the most attractive candidates: Additive manufacturing guarantees the low cost of PE, while different material and substrate choices allow for the realization of non-toxic and flexible circuits [4]. Compared to silicon technology, PE cannot exceed in terms of performance, integration density, and area. Therefore, rather than replacing silicon-based electronics, PE serves as a complement of them.

Neuromorphic computing refers mainly to the combination of weighted-sum operation and non-linear activation inspired by synaptic neurons in the human brain. Despite the simplicity of its primitive operations, it has shown to have strong (non-linear) expressiveness [5] and has achieved extraordinary success in many fields [6].

Bringing together the advantages of both PE and neuromorphic computing, printed neuromorphic circuits were proposed (e.g., [7]). In printed neuromorphic circuits, the weighted-sum operation and the activation function are implemented by crossbar and non-linear circuitry. Further, similar to the neural networks in silico, by interconnecting multiple printed neurons, a circuit capable for expressing more sophisticated functionality can be realized. Moreover, machine learning based processes can be applied to printed neuromorphic circuits, which ensures the high efficiency of their designing. Based on these advantages, printed neuromorphic circuits become highly competitive in the aforementioned emerging domains.

The production of PE can be categorized into two types: high-volume production, such as gravure- and screen-printing [8], and low-volume or customized manufacturing, like jet-printing [9]. High-volume manufacturing generally leads to lower individual costs and production time of each circuit, but allows to only produce replications of a single template. Conversely, low-volume manufacturing provides maximum flexibility, but the cost-per-circuitry becomes higher, and the production time is increased.

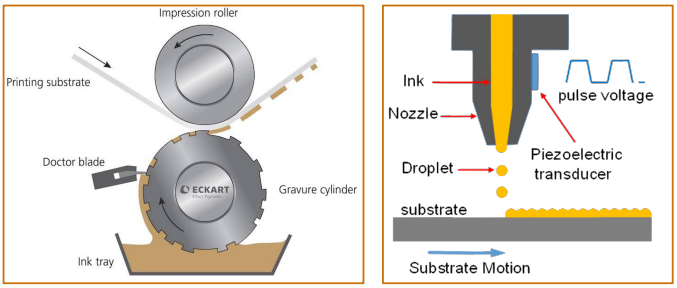
On this premise, in case numerous circuits for diverse tasks need to be printed, low-volume printing technologies are usually employed, as circuits for different tasks usually do not share components. Thus, the production cost could be high. Alternatively, for cost saving, same circuits can be printed by high-volume printing processes to address all the tasks, but the performance (e.g., accuracy in classification tasks) could be worse. In this regard, neither high- nor low-volume production can consider both cost and performance.

In this work, we leverage the additive manufacturing feature of PE to bridge the gap between high- and low-volume printing technologies by combining high-throughput (high-volume) printing with point-of-use (low-volume) customization. This allows for a reduction of the manufacturing costs, while retaining an acceptable circuit performance.

Ideally, different (neuromorphic) circuits, while fulfilling different tasks, display high commonality in their designs. In this case, a large subset of the circuit components may be fabricated in an initial, shared, high-volume production step. Afterwards, point-of-use additive post-printing adjustment can be applied to configure the printed circuits to specific tasks. To achieve this, we propose a common training (design) strategy for multiple printed neuromorphic circuits. Through this strategy, we aim to encourage high commonalities in printed neuromorphic circuits and thus minimize the required post-printing effort. We refer to this as *split additive manufacturing*.

In summary, the contributions of this work are:

- We formulate the split additive manufacturing as well as the physical constraints mathematically. Based on this formulation, we propose the super printed neural network, a model to train (design) multiple neuromorphic circuits for different tasks simultaneously, while encouraging high design similarities.



Gravure Printing **Inkjet Printing**

Fig. 1. Schematic of different printing technologies, gravure printing (left) and inkjet-printing (right).

- We investigate the relationship between the adjustment of individual configurations and the circuits' performance. The trade-off can be described by a Pareto curve.
- We train a super printed neural network on 30 benchmark datasets (simultaneously) to evaluate the capabilities of our approach of co-designing printed neuromorphic circuits.

The preliminary experiment shows that, 61.4% point-of-use printing cost can be saved without loss in accuracy. In case a 10% drop in accuracy were allowed, 94.3% of this cost could be saved.

The rest of this paper is structured as follows: Sec. II introduces PE and printed neural network (pNN). Sec. III describes the formulation of the super pNN, as well as the technique to reduce the cost for individual configuration. In Sec. IV, the proposed approach is evaluated and discussed. Finally, Sec. V concludes this work and discusses possible future works.

II. Preliminary

A. Printed Electronics

Printed electronic (PE) technology refers to manufacturing that is based on various printing processes, such as gravure-printing and jet-printing. Due to the simplicity of the manufacturing process and the low cost of equipments, printed circuits can be fabricated with significantly lower cost compared to silicon-based electronics. Moreover, PE provides the option to print on flexible substrates, such as Kapton [4] or PET [10], so that the produced devices can also be flexible. Rather than replacing silicon-based electronics, PE serves as a complement since it cannot precede silicon-based chips regarding performance, integration density, and area. Therefore, PE has a high compatibility in many emerging domains like wearable computing and near-sensor computing.

Regarding the scale of production, printing technologies are broadly divided into two categories. Replication printing technologies are designed for high-volume printing, such as gravure- and screen-printing [8]. Gravure-printing (see Fig. 1, left) is a type of intaglio printing process. In gravure-printing, the (circuit) pattern will be engraved onto a cylinder carrier, after which a large amount of circuits can be fabricated by rotating the cylinder carrier. Screen-printing refers to hollowing out (circuit) patterns on a stencil, and using a squeegee to press the printing material onto the substrate in the hollowed out areas. In these two high-volume processes, the masks (cylinder carrier or stencil) are hard to be modified once they are defined, however, the masks can be reused repeatedly and allow for cheap and fast production. Contrary to replication printing technologies, jet-printing technologies like inkjet-printing (see Fig. 1, right) are more appropriate for individual manufacturing. By adapting the printing route and material for each task, circuitry with a broad spectrum of functionalities can be achieved. Unfortunately, the functionality of individual printing comes at a price of higher production times and higher (variable) costs per circuit. This technology should therefore only be employed when the flexibility or custom fabrication capabilities are required.

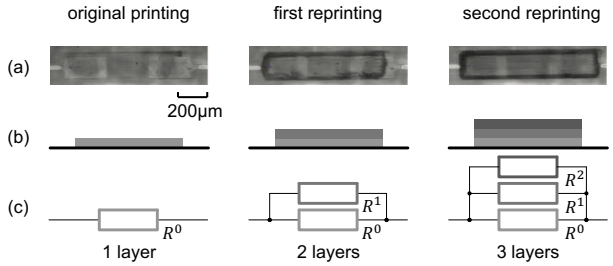


Fig. 2. Resistor reprinting by adding layers. (a) microscope photos, (b) physical schematics, (c) circuit diagrams.

B. Reprinting Resistors in Printed Electronics

Thanks to the additive manufacturing characteristic of PE, printed circuits can be adjusted post-fabrication. This is possible by selectively adding material or modifying the geometric shape of a component. Fig. 2 shows a printed resistor with additional layers of conductive ink added post-fabrication to adjust its conductivity. This additive characteristic enables to combine multiple printing technologies and bridge the gap between them. For brevity, this procedure will be referred to as *reprinting* in the following. Note that, as can be seen in Fig. 2, reprinting cannot adjust the existing resistance arbitrarily, but only decrease it. This is because adding additional conductive (resistive) material creates parallel resistive structures, which lead to lower overall resistance. Hence, reprinting can only decrease resistance.

C. Printed Neural Network

Similar to classical artificial neural networks, printed neuromorphic circuits are composed of printed neurons. Fig. 3 shows an exemplary printed neuromorphic circuit and an enlarged schematic of a printed neuron. According to Kirchhoff's law [11], we obtain

$$V_z g_d = \sum_i (V_i - V_z)g_1 + (V_b - V_z)g_b; \qquad (1)$$

where V_1, V_2, ..., V_b denote the input voltages and V_z refers to the output voltage of the crossbar (green part in Fig. 3) and $g_i = 1/R_i$ denotes the conductance of a resistor R_i. After summarizing $G = \sum_i g$, Eq. 1 can be formulated as

$$V_z = \sum_i \frac{g_i}{G} V_i + \frac{g_b}{G} V_b.$$

By interpreting $w_i = g_i/G$, the similarity to the weighted-sum operation in a neural network, $\sum_i w_i x_i + b$, becomes

Fig. 3. An exemplary neural network (left) and an enlarged schematic of a printed neuron (right). The green part indicates the crossbar for weighted-sum operation, and the red part denotes a tanh-like activation function circuit.

evident, i.e., the weights in neural networks are embodied in the conductances of the printed resistors. By connecting a printed tanh-like activation function circuit (pink part in Fig. 3) to the crossbar, circuitry resembling a printed neuron can be implemented.

Generally, only the conductance values g of the crossbar resistors are modified, as they correspond to the weights in the neural network. To find suitable values for these conductances and solve, e.g., a classification task, a pNN (mathematical model of the printed neuromorphic circuit) is generated. The learnable parameters of the pNN are the surrogate conductances θ that resemble the conductance values g and the corresponding negative weight circuits [7]. The negative weight circuit is designed to emulate negative weights. This is necessary, as printed conductance can only be positive, therefore, when the learned weight w_i (surrogate conductance θ) is negative, the conductance $g_i = |\theta_i|$ should be printed and the corresponding input voltage is inverted ($inv(V_i)$) through the negative weight circuit. In other words, the absolute value of the surrogate conductance $|\theta_i|$ indicates the conductance to be printed, while the sign of θ_i denotes whether the input voltage V_i should be inverted. In training, surrogate conductances are obtained via gradient-based learning. In this way, training the pNN may be understood as designing a printed neuromorphic circuit.

III. Methodology

PE involves both high- and low-volume fabrication processes. However, neither can cover both circuit performances (e.g, classification accuracy) and printing costs simultaneously, in case numerous circuits for diverse tasks need to be printed. In this section, we bridge the gap between two types of technologies by leveraging the reprinting technology, and aim to achieve a higher efficiency for the fabrication with acceptable circuit performances.

For this, we try to find common parts in a set of printed neuromorphic circuits. These common parts can be fabricated via a high-volume process. Then, each individual circuit is customized via point-of-use reprinting. Through this, the costs and time of individual printing are reduced, while retaining an acceptable accuracy. For this purpose, we introduce the super pNN, a model for training multiple pNNs for different tasks simultaneously. The super pNN considers a decomposition of the conductances of multiple pNNs into a common part, which is shared across all pNNs, and individual parts, which vary among the pNNs of different tasks. By fabricating the circuits with combined technologies, costs and production time can be saved, while attaining comparable performances for the circuits.

Fig. 4. Structure of learnable parameters in an exemplary super pNN. Each pNN has its own individual conductance θ^I_t, while the common conductance θ^C is shared across all pNNs. The resultant conductance of the pNN for the t-th task θ_t equals $\theta^C + \theta^I_t$ and determines the resulting weights in the pNN. The input/output layers of all pNNs in a super pNN are padded to the same dimensionality. The inputs denoted by 0 will be connected to ground.

A. Conductance Splitting

As illustrated in Fig. 2, reprinting a resistor, i.e., printing additional material on top of the existing resistor can be seen as printing an additional, parallel conductive path. Hence, the total conductance of the reprinted resistor is the sum of the conductances of the old conductive path and that of the newly added geometry. The conductance value of a resistor produced via an initial fabrication step (high-volume) and later individually customized via, e.g., inkjet-printing, can thus be expressed as $\theta = \theta^C + \theta^I$. Here, θ^C (common surrogate conductance) denotes the conductance value of the initially printed device. As it is fabricated with a high-volume process, it is shared by all circuits fabricated with the same mask. Consequently, θ^I (individual surrogate conductance) denotes the conductance value of the additional material printed post-fabrication to customize the device in a reprinting step. In the following, we refer to the vector θ as the summary of all conductance values θ_i in a pNN, i.e., $\theta = [\theta_1, \theta_2, \ldots]$. Analogously, θ^C and θ^I summarize their corresponding common and individual conductance values.

B. Super pNN

When pNNs should be fabricated for a set of different tasks $t = 1, \ldots, T$, we need to design/train T different pNNs to address them. If these pNNs are trained independently, little commonality can be expected between them. Thus, there is little potential for joint production and split manufacturing. To increase this potential, the training of these pNNs should be done jointly. For this purpose, we introduce the super pNN. A super pNN is a model for training pNNs of different tasks jointly to achieve a high commonality and thus high potential for joint production via split additive manufacturing. For a set of T tasks, a super pNN has a set of parameters θ^C, which is shared among all pNNs, and several sets of parameters θ^I_t with $t = 1, \ldots, T$ that are unique to each individual task

TABLE I
BENCHMARK DATASETS AND BASELINE ACCURACY

dataset	# input # output	# data	baseline accuracy
Acute In ammation	6-2	120	0.904
Acute Nephritis	6-2	120	0.925
Balance Scale	4-3	625	0.671
Blood	4-2	748	0.747
Breast Cancer	9-2	286	0.724
Breast Cancer Wisconsin	9-2	699	0.955
Breast Tissue	9-6	106	0.409
Ecoli	7-8	336	0.592
Energy (y_1)	8-3	768	0.816
Energy (y_2)	8-3	768	0.761
Fertility	9-2	100	0.857
Glass Identi cation	9-6	214	0.439
Haberman's Survival	3-2	306	0.788
Hayes-Roth	3-3	132	0.342
ILPD	9-2	583	0.684
Iris	4-3	150	0.701
Mammographic Mass	5-2	961	0.728
Monks-1	6-2	124	0.598
Monks-2	6-2	169	0.617
Monks-3	6-2	122	0.576
Pima Indians Diabetes	8-2	768	0.644
Pittsburgh Bridges MATERIAL	7-3	106	0.902
Pittsburgh Bridges SPAN	7-3	92	0.509
Pittsburgh Bridges T-OR-D	7-2	102	0.800
Pittsburgh Bridges TYPE	7-6	105	0.665
Seeds	7-3	210	0.454
Teaching Assistant Evaluation	5-3	151	0.397
Tic-Tac-Toe Endgame	9-2	958	0.632
Vertebral Column (2 cl.)	6-2	310	0.635
Vertebral Column (3 cl.)	6-3	310	0.586

The (surrogate) conductance vector of a pNN trained for task t is then implicitly determined by $_t := ^C + ^I_t$. A conceptual illustration can be seen in Fig. 4. Naturally, for this to work, the topologies of all pNNs need to be compatible in the sense that they have the same number of input, hidden, and output neurons. To address this, we take the maximal number of input/output among all pNNs as the number of input/output for all pNNs (see Fig. 4). For tasks with fewer inputs, zero-padding is employed in training, which relates to connecting those inputs to 0V (GND) in the circuits. Consequently, irrelevant outputs can simply be ignored.

To achieve valid pNNs, several constraints of the printing technology need to be respected. Firstly, the range of printable conductance values is given by $0 \cup [\,[G_{min}; G_{max}]$, where G_{max} and G_{min} depend on the speci c technology and $= 0$ refers to no printing. These constraints must hold for both $_j$ and the $j\,^I_t j$, and can be addressed via projections in training, see [7], and the use of the straight-through estimator as in [12].

Beyond that, we also have to consider that reprinting can only increase the conductances from their original values, i.e., $j\,_t j \;\; j\,^C j$ for any task t. In other words, we cannot change the choice of connecting either V_i or $\mathrm{inv}(V_i)$, to adjust what would relate to the sign of w_i via reprinting. To respect this constraint, ^{C} determines the signs of the entries of $_t$ via

$$_t := \mathrm{sign}(\,^C) \;\; j\,^I_t j + \;^C;$$

while ^I_t is only able to adjust the absolute value of the resulting conductances.

C. Training Objective

Through the super pNN, a connection between different tasks is established via the common ^{C}. However, the de nition of the super pNN does not necessitate high commonality between the individual pNNs. For example, the solution after training may likely have $^C = 0$ and express everything via the uncoupled ^I_t. Therefore, to encourage high commonality, we add a penalty term to the training objective to keep the individual conductances ^I_t, and thus the reprinting effort, low. In this work, the penalty term is formulated as the $_1$-norm of all the individual conductances ^I_t, i.e.,

$$C(\,^I) = \sum_t \;\; ^I_t \;\; _1; \qquad (2)$$

because both printing times and the amount of the printing materials of the individual printing are approximately proportional to the size of the entries of ^I_t. Consequently, the training objective of the super pNN considering both accuracy and reprinting costs is then given by

$$L(\,^C; \,^I) = \sum_t L(\,^C; \,^I_t; x_t; y_t) + \;\; C(\,^I); \qquad (3)$$

where $L(\,)$ denotes the loss function proposed in [7], which takes hardware-related constraints into account. The vectors x_t and y_t are the training inputs and labels of the t-th task. Furthermore, the coef cient $\;\; 0$ denotes a (hyper-parameter) adjusting the in uence of the costs $C(\,^I)$.

For $= 0$, the training objective is unaffected by the cost $C(\,^I)$ of the reprinting required for each task. In this case, the training of the super pNN can be conceptually equated to the completely independent training of T pNNs for T tasks. Consequently, there may be little commonality between the different pNNs and thus little potential for a sensible production of ^{C} via a high-volume production process. However, the individual pNNs may also achieve the best accuracy as they are not bound together. On the other hand, for high values of $\;\;$ (potentially $\;\; ! 1$, in practice greater than a certain threshold), the cost $C(\,^I)$ will completely dominate the loss term in training. This should lead to a solution where $8t : \,^I_t = 0$. In this case, the individual pNNs are solely determined by ^{C}, and are thus all the same. While this is the most economical in terms of production costs, the resulting pNNs likely provide no useful accuracy for their respective tasks. Thus, may be used to express a trade-off between cost and accuracy. To nd an appropriate value of , we suggest training the super pNN for different values of and draw a Pareto front [13].

IV. EVALUATION

We implement[1] the proposed super pNN with PyTorch, and conduct an experiment with 30 benchmark datasets, whose complexities and use cases match the PE and pNN pro le. The results are additionally analyzed regarding the accuracy-cost trade-off by generating a Pareto front of possible solutions.

[1] https://github.com/Neuromophic/SpLM_Manufacturing_One_Mask

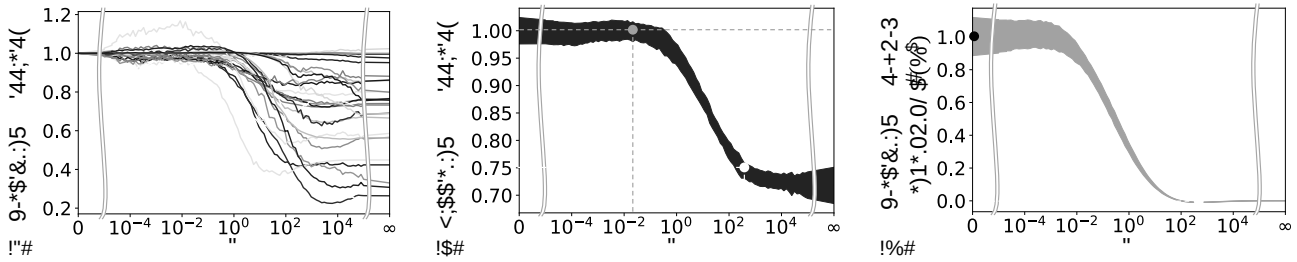


Fig. 5. Results of experiment with 100 different values. (a) normalized accuracies of 30 tasks. Each task is indicated by a different color. (b) summarized accuracies (average normalized accuracies), the blue curve and area denote the mean and standard deviation respectively. (c) normalized cost of the individual (point-of-use) reprinting, the red curve and area denote the mean and standard deviation respectively.

A. Experiment

a) Datasets: For our experiments, a subset of the 121 classification benchmark datasets summarized in [14] was taken. We select datasets which are suitable for PE and printed neuromorphic circuits, most notably, tasks with a limited number of inputs and outputs (≤ 10). Moreover, since numerous pNNs are trained simultaneously, we limit the experiments to datasets with the number of data points between 100 and 1 000, which leaves 30 datasets. Finally, we scale all the inputs to $[0; 1]$ to simulate the electrical signals from sensors and put all inputs on the same scale. We split each dataset into training (60%), validation (20%), and test (20%) sets. The detailed information about the datasets can be found in Tab. I.

b) Hyper-parameters: As described in Sec. III-B, the topology of all pNNs is determined by the maximal number of input and output of all the datasets. In this experiment, the topology is chosen to be 9-3-8, where the number of inputs and outputs are determined by the datasets. For training, we use the Adam [15] optimizer with default parameters and a learning rate of 0:01. As the stop criterion, we employ the early-stopping with 1000-epoch patience on validation loss. In investigate trade-off between accuracy and cost, we select $\in [10^{-5}; 10^{5}]$ equidistant in log space. To generate the conceptual boundary cases of $= 0$ and $\to \infty$, we train only with ^{l} (for $= 0$) and ^{c} (for $\to \infty$).

The training is repeated 30 times (with seeds varying from 1 to 30) for different initialization for each value of to make sure to achieve a sufficiently good solution for each value of . Finally, all the hardware-related hyper-parameters, such as measuring threshold and margin, are taken from [12].

c) Baseline: As the baseline, we report the performance of the super pNN corresponding to $= 0$, which equivalent to train the super pNN considering only individual ^{l}. This result can be regarded as the upper bound of the circuit performance. We also refer to this as *individual* pNNs.

B. Result

After training, we evaluate the super pNNs on the test sets. Tab. I reports the accuracies of baseline. To analyze the impact of more clearly and to eliminate the disparate difficulties among different tasks, we normalize the accuracy by the baseline, which should theoretically be able to achieve the best results. Note that this is not always achieved in practice due to the complex nature of the nonlinear optimization problem that neural network training resembles. The resulting curves are displayed in Fig. 5(a).

To summarize the overall implications of for super pNN, we report the *summarized accuracy*, which refers to the average value of the normalized accuracies over all the tasks in a super pNN. The result of all runs is shown in Fig. 5(b). We also show the relationship between and the cost of reprinting in Fig. 5(c). To obtain the Pareto front, we plot all the $C(\ ^{l})$ versus their summarized accuracy for all runs and all values of in Fig. 6. Based on the scatters, we draw the Pareto front as the red dashed line.

C. Discussion

As can be seen in Fig. 5, at $= 0$, which refers to fully individual printing, the summarized accuracy is normalized to 1, and the expected cost of individual reprinting is the highest. Analogous to the treatment of the accuracy, we normalize the cost by this value, as shown by the black point in Fig. 5(c), where $C(\ ^{l})$ denotes the normalized reprinting cost. With increasing , the normalized accuracy, summarized accuracy, and the cost of reprinting $C(\ ^{l})$ will decrease. For greater than a certain threshold, i.e., $\geq 10^{2:5}$ in this experiment, the cost penalty completely dominates the training objective. Thus, as expected, $^{l}_{t} = 0$ for all pNNs, and the training results are equivalent to performing all tasks with the same pNNs. We refer to this pNN as *common* pNN. The orange points in Fig. 5(b), (c) and Fig. 6 show the common pNN with the best accuracy.

Contrary to our expectation, the summarized accuracy does not decrease immediately as is increased from 0. This could be due to two possible reasons. Firstly, the $C(\ ^{l})$ term may act as a regularization and mitigates overfitting to the training set of the specific task. Therefore, the accuracy of some pNNs could even slightly increase, e.g., the pink curve in Fig. 5(a). Secondly, only explicitly affects ^{l} rather than ^{c}. Hence, ^{c} would be adjusted accordingly during training to compensate for the loss of accuracy caused by the penalty term. This phenomenon allows reducing the cost of individual printing, while retaining an acceptable accuracy. Such a solution can be found at the purple point in Fig. 5(b), where $\approx 10^{-1:5}$.

The Pareto curve in Fig. 6 reveals the relationship between the cost for individual configuration and the summarized accuracies of super pNNs. Compared to fully individual printing (black point), the point-of-use printing cost can be reduced to 38:6% without any noticeable loss in accuracy (purple point).

TABLE II
ACCURACY-COST TRADE-OFF

	summarized accuracy	normalized cost of reprinting $C(\ ^{l})$
individual	100%	100.0%
split additive manufacturing	100%	38.6%
	95%	15.7%
	90%	5.7%
	85%	2.9%
	80%	1.4%
common	75%	0.0%

Moreover, if the summarized accuracy is allowed to reduced by e.g., 5%, the reprinting cost can be further reduced to 15.7% (blue point). Finally, using a single *common* pNN (orange point) leads to a 75% summarized accuracy, but consequently also no reprinting costs. Other exemplary trade-off options are summarized in Tab. II.

In other word, based on the (common) throughput printing, the accuracy increases rapidly as the allowed cost for individual printing increases and then saturates. In other words, relatively little adjustment from the joint configuration of ^{C} of the individual conductances $^{l}_{t}$ leads to a significant improvement in performance for the individual pNNs. Thus, there is great potential for a joint fabrication of multiple pNNs via split additive manufacturing.

V. CONCLUSION AND FUTURE WORK

In this work, we propose a design strategy for multiple different printed neuromorphic circuits to leverage split additive manufacturing. We combine and bridge the gap between high- and low-volume printing. The experiment shows that, given pNNs that are trained together (co-designed) with shared conductances in our proposed super pNN model, a substantial amount of point-of-use printing cost can be saved, while still obtaining pNNs with sufficient accuracies. This may result in substantial time savings when fabricating printed neuromorphic circuits. Also, depending on the requirements, manufacturers may be able to adjust their production strategy based on the cost-accuracy trade-off visualized by the Pareto front.

Despite the preliminary progress made in this work, many possibilities remain for future work. First, the performances of some tasks drop dramatically with increasing , i.e., these tasks are potentially not suitable to use the common conductance with other tasks. This problem can be addressed by either fabricating them individually, or by creating groups of tasks for joint fabrication dynamically. Additionally, for this work, we chose to estimate the costs of reprinting $C(\ ^{l})$ by the sum of the $`_1$ norms of the $^{l}_{t}$. However, this cost would depend also on, e.g., the distribution of $^{l}_{t}$ at each individual position, the printing technique adopted, etc. Therefore, a more elaborate cost model could be developed that considers reprinting capabilities and efforts more concretely. Lastly, the $C(\ ^{l})$ is controlled implicitly by . Future work could allow constraining the costs explicitly, so that the optimal performance for a given budget can be found.

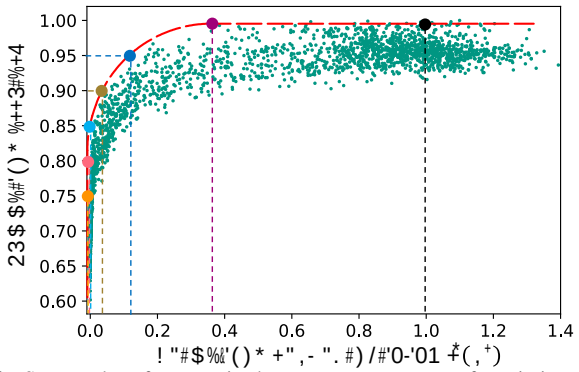


Fig. 6. Scatter plot of summarized accuracy versus cost of reprinting for all the runs. The red curve displays the Pareto front and the bold points denote different possible trade-offs on the Pareto front.

ACKNOWLEDGMENT

This work has been partially supported by the Carl-Zeiss-Foundation as part of "stay young with robots" (Jubot) project.

REFERENCES

[1] H. Zhao, A. Scholz, M. Beigl, S. Ni, S. A. Singaraju, and J. Aghassi-Hagmann, "Printed Electrodermal Activity Sensor with Optimized Filter for Stress Detection," in *International Symposium on Wearable Computers (ISWC'22), Atlanta, GA and Cambridge, UK, September 11-15, 2022*.

[2] F. Zhou and Y. Chai, "Near-sensor and In-sensor Computing," *Nature Electronics*, vol. 3, no. 11, pp. 664–671, 2020.

[3] F. Liu, G. Tang, Y. Li, Z. Cai, X. Zhang, and T. Zhou, "A Survey on Edge Computing Systems and Tools," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1537–1562, 2019.

[4] I. I. Labiano and A. Alomainy, "Flexible Inkjet-printed Graphene Antenna on Kapton," *Flexible and Printed Electronics*, vol. 6, no. 2, p. 025010, 2021.

[5] Z. Yu, A. M. Abdulghani, A. Zahid, H. Heidari, M. A. Imran, and Q. H. Abbasi, "An Overview of Neuromorphic Computing for Artificial Intelligence Enabled Hardware-based Hopfield Neural Network," *IEEE Access*, vol. 8, pp. 67 085–67 099, 2020.

[6] C. D. Schuman, T. E. Potok, R. M. Patton, J. D. Birdwell, M. E. Dean, G. S. Rose, and J. S. Plank, "A Survey of Neuromorphic Computing and Neural Networks in Hardware," *arXiv preprint arXiv:1705.06963*, 2017.

[7] D. D. Weller, M. Hefenbrock, M. Beigl, J. Aghassi-Hagmann, and M. B. Tahoori, "Realization and Training of an Inverter-based Printed Neuromorphic Computing System," *Scientific reports*, vol. 11, no. 1, pp. 1–13, 2021.

[8] Z. Cui, *Printed Electronics: Materials, Technologies and Applications*. John Wiley & Sons, 2016.

[9] Z. Yin, Y. Huang, N. Bu, X. Wang, and Y. Xiong, "Inkjet Printing for Flexible Electronics: Materials, Processes and Equipments," *Chinese Science Bulletin*, vol. 55, no. 30, pp. 3383–3407, 2010.

[10] J. Chang, T. Ge, and E. Sanchez-Sinencio, "Challenges of Printed Electronics on Flexible Substrates," in *2012 IEEE 55th international midwest symposium on circuits and systems (MWSCAS)*. IEEE, 2012, pp. 582–585.

[11] G. Kirchhoff, "LXIV. On a Deduction of Ohm's Laws, in connexion with the Theory of Electro-statics," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 37, no. 252, pp. 463–468, 1850.

[12] H. Zhao, M. Hefenbrock, M. Beigl, and M. B. Tahoori, "Aging-Aware Training for Printed Neuromorphic Circuits," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD '22)*, 2022.

[13] I. Giagkiozis and P. J. Fleming, "Pareto Front Estimation for Decision Making," *Evolutionary computation*, vol. 22, no. 4, pp. 651–678, 2014.

[14] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do We Need Hundreds of Classifiers to Solve Real World Classification Problems?" *The journal of machine learning research*, vol. 15, no. 1, pp. 3133–3181, 2014.

[15] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014.