Transformer Network-Based Reinforcement Learning Method for Power Distribution Network (PDN) Optimization of High Bandwidth Memory (HBM)

Hyunwook Park, Graduate Student Member, IEEE, Minsu Kim, Seongguk Kim, Graduate Student Member, IEEE, Keunwoo Kim, Graduate Student Member, IEEE, Haeyeon Kim, Taein Shin, Keeyoung Son, Graduate Student Member, IEEE, Boogyo Sim, Graduate Student Member, IEEE, Subin Kim, Seungtaek Jeong, Member, IEEE, Chulsoon Hwang, Senior Member, IEEE, and Joungho Kim, Fellow, IEEE

Abstract—In this first article, for the first time, we propose a transformer network-based reinforcement learning (RL) method for power distribution network (PDN) optimization of high bandwidth memory (HBM). The proposed method can provide an optimal decoupling capacitor (decap) design to maximize the reduction of PDN self- and transfer impedances seen at multiple ports. An attention-based transformer network is implemented to directly parameterize decap optimization policy. The optimality performance is significantly improved since the attention mechanism has powerful expression to explore massive combinatorial space for decap assignments. Moreover, it can capture sequential relationships between the decap assignments. The computing time for optimization is dramatically reduced due to the reusable network on the positions of probing ports and decap assignment candidates. This is because the transformer network has a context embedding process to capture meta-features including probing ports positions. In addition, the network is trained with randomly generated datasets. The computing time for training and data cost are critically decreased due to the scalability of the network. Due to its shared weight property and the context embedding process, the network can adapt to a larger scale of problems without additional training. For verification, the results are compared with conventional genetic algorithm (GA), random search (RS), and all the previous RL-based methods. As a result, the proposed method outperforms in all the following aspects: optimality performance, computing time, and data efficiency.

Index Terms—Decoupling capacitor (decap), high bandwidth memory (HBM), power distribution network (PDN), reinforcement learning (RL), transformer network.

I. INTRODUCTION

RECENTLY, artificial intelligence (AI) era requires over the terabyte-per-second (TB/s) bandwidth between the processing units and memories in the AI server [1], [2]. To meet the demands, a graphic processing unit (GPU)–high bandwidth memory (HBM) module has been regarded as a promising solution. The GPU–HBM provides over several TB/s bandwidth between GPU and HBM with 1024 I/Os [3], [4]. This TB/s scale bandwidth is possible due to the silicon interposer and through silicon via (TSV) technologies [5], [6]. Those enable fine pitch interconnections and high density in the I/O interface. However, due to the increasing data rate from 1 Gb/s at gen 1 to 6.4 Gb/s at gen 3 and the tremendous number of I/Os, switching power in the HBM I/O interface has been increased by generations [7], [8]. Moreover, the supply voltage level keeps shrinking to limit the power consumption [3]. Therefore, robust power distribution network (PDN) design becomes further challenging. It is essential to ensure signal quality by lowering power supply noise (PSN) fluctuation and power supply induced jitter (PSIJ) [9], [10], [11].

Decoupling capacitor (decap) design is one of the most important processes to suppress PSN and PSIJ by lowering PDN impedance [12]. Simultaneous switching noise (SSN) is the main noise source in the HBM I/O interface, which leads to severe eye distortion, as shown in Fig. 1(a) [13], [14]. Large SSN can be generated when the simultaneous switching current drawn by I/O buffers meets the high peak anti-resonances of the VDDQ PDN. Therefore, it is necessary
to optimize decap design to lower the PDN impedance in the broadband frequency range while minimizing the layout area to reduce the process cost as well as the power consumption [15], [16], [17]. However, decap design optimization is a combinatorial optimization problem that has high computational complexity. This is because the previous decap assignment affects the next assignment in terms of PDN impedance reduction [18]. A large scale of the PDN decap optimization requires extremely high computing cost, because the size of PDN Z-matrix to be optimized is increased proportional to the square of the number of ports and proportional to the number of frequency points and dataset size. In addition, the decap optimization requires high computing time since the computation of the PDN impedance is time-consuming. Also, massive iterations are needed for the optimization. In that point of view, the equivalent circuit modeling of the PDN should be preceded for the optimization, rather by the 3-D electromagnetic (EM) simulator [13]. Furthermore, a computing time and cost-effective equivalent circuit modeling of the PDN should be preceded for the optimization. In that point of view, the PDN impedance is time-consuming. Also, massive iterations requires high computing time since the computation of the points and dataset size. In addition, the decap optimization number of ports and proportional to the number of frequency scale of the PDN decap optimization requires extremely high assignment in terms of PDN impedance reduction [18]. A large This is because the previous decap assignment affects the next assignment for the design metric. Those exclude transfer impedance, which indicates the coupled PSN [21], [22], [23], [24], [25], [26], [27], [28]. Moreover, all the previous RL-based optimization methods using a multilayer perceptron (MLP) as the value function approximator are investigated [23], [25]. Also, on-interposer and on-chip decap optimization using deep Q learning with a convolutional neural network (CNN) is studied [24]. However, those are limited in both the size of the solution space and the optimality performance, because they are based on the value-based RL algorithms with simple value approximators such as the MLP and CNN. To increase the solution space size, direct policy parameterization by policy networks and training the networks through policy gradient algorithms are investigated [26], [27]. In addition, Park et al. [27] introduced a transformer network to consider sequential relationships between the decap assignments for performance improvement. However, both of them do not have the reusability causing the increment of the optimization time. A recent study proposed a reusable method on the position of the probing port [28]. However, all the previous RL-based methods including [28] are limited in scalability, which causes the increment of computing time and cost when training [21], [22], [23], [24], [25], [26], [27], [28]. The scalability refers to the characteristic of whether the trained neural network model can respond to the scale of the problem. Moreover, all the previous RL-based methods only considered self-impedance for the design metric. Those exclude transfer impedance, which indicates the coupled PSN [21], [22], [23], [24], [25], [26], [27], [28]. In addition, multiprobing ports are not considered [21], [22], [23], [24], [25], [27], [28].

In this article, we propose a transformer network-based RL method for PDN decap optimization of the HBM. The proposed method can optimize decap design to maximize the reduction of both the self- and transfer impedances seen at the multiple ports. Consideration of the sequential relationships between the decap assignments using the transformer network achieves higher optimality performance. Moreover, the network has a context embedding process to capture meta-feature including positions of probing ports. The context embedding process increases the generalization ability to new problems. Therefore, the network trained with randomly generated datasets is reusable to new problems varying positions of probing ports and decap candidates. Due to the shared
weight property and the context embedding, the network has the generalization ability on the scale of decap optimization problems. The reusability and scalability provide a significant reduction of the computing time and cost for both the training and optimization. By comparing with the conventional GA and RS methods and all the previous RL-based methods, we validate the higher optimality and the lower computing cost and time.

II. PROPOSAL OF TRANSFORMER NETWORK-BASED RL METHOD FOR PDN DECAP OPTIMIZATION

In this section, a transformer network-based RL method for the PDN decap optimization is proposed. A brief explanation of the decap optimization problem for this work is as follows. It is denoted as a decap n/m problem. The problem formulation of the decap n/m is as follows: assigning m number of unit NMOS decaps for given n number of positions for decap assignment candidates X to maximize the reduction of the self- and transfer impedances seen at four probing ports P. X is a set of feature vectors of n decap ports from on-chip and on-interposer HBM VDDQ PDNs

\[ X = \{x_1, x_2, \ldots, x_n\}. \tag{1} \]

P is a set of four probing ports in the I/O region of a physical layer (PHY) of the on-chip VDDQ PDN

\[ P = \{x_{p1}, x_{p2}, x_{p3}, x_{p4}\}. \tag{2} \]

Therefore, a total ten number of the self- and transfer impedances become a decap design criterion: \(Z_{11}, Z_{22}, Z_{33}, Z_{44}, Z_{12}, Z_{13}, Z_{14}, Z_{23}, Z_{24}, Z_{34}\). The total SSN is the sum of the self-noise and transfer noises [5]. The amount of self-noises is related to the self-impedances and that of the transfer noises is related to the transfer impedances. Thus, all the ten impedances need to be considered for the decap design criterion.

Fig. 2 shows the overall concept of the proposed transformer network-based RL method for the PDN decap optimization. For a given input state \(s\), a policy network outputs a decap assignment sequence \(a\). The MDP parameters are defined as follows: state \(s\) is a set of feature vectors of the probing ports \(x_{p1:p4}\) and the decap assignment candidates \(X\) to maximize the reduction of the self- and transfer impedances seen at four probing ports \(P\). X is a set of feature vectors of \(n\) decap ports from on-chip and on-interposer HBM VDDQ PDNs

\[ X = \{x_1, x_2, \ldots, x_n\}. \tag{1} \]

P is a set of four probing ports in the I/O region of a physical layer (PHY) of the on-chip VDDQ PDN

\[ P = \{x_{p1}, x_{p2}, x_{p3}, x_{p4}\}. \tag{2} \]

Therefore, a total ten number of the self- and transfer impedances become a decap design criterion: \(Z_{11}, Z_{22}, Z_{33}, Z_{44}, Z_{12}, Z_{13}, Z_{14}, Z_{23}, Z_{24}, Z_{34}\). The total SSN is the sum of the self-noise and transfer noises [5]. The amount of self-noises is related to the self-impedances and that of the transfer noises is related to the transfer impedances. Thus, all the ten impedances need to be considered for the decap design criterion.

Fig. 2 shows the overall concept of the proposed transformer network-based RL method for the PDN decap optimization. For a given input state \(s\), a policy network outputs a decap assignment sequence \(a\). The MDP parameters are defined as follows: state \(s\) is a set of feature vectors of the probing ports \(x_{p1:p4}\) and the decap assignment candidates \(X\) to maximize the reduction of the self- and transfer impedances seen at four probing ports \(P\). X is a set of feature vectors of \(n\) decap ports from on-chip and on-interposer HBM VDDQ PDNs

\[ X = \{x_1, x_2, \ldots, x_n\}. \tag{1} \]

P is a set of four probing ports in the I/O region of a physical layer (PHY) of the on-chip VDDQ PDN

\[ P = \{x_{p1}, x_{p2}, x_{p3}, x_{p4}\}. \tag{2} \]

Therefore, a total ten number of the self- and transfer impedances become a decap design criterion: \(Z_{11}, Z_{22}, Z_{33}, Z_{44}, Z_{12}, Z_{13}, Z_{14}, Z_{23}, Z_{24}, Z_{34}\). The total SSN is the sum of the self-noise and transfer noises [5]. The amount of self-noises is related to the self-impedances and that of the transfer noises is related to the transfer impedances. Thus, all the ten impedances need to be considered for the decap design criterion.

The context embedding process increases not only the optimality but also the generalization ability to new problems varying locations of \(P\) and the decap candidates \(X\). Details on the context embedding are explained in Section II-B. Moreover, the policy network is trained with randomly generated datasets for reusability. The datasets are the states and corresponding PDN Z-matrices, with randomly generated locations for \(P\) and \(X\). Details on the randomly generated data are described in Section II-D. The computing time for the RL-based methods is composed of training time to train the network and optimization time to obtain the optimized solutions. The reusability can reduce the optimization time because only one inference and one reward estimation are required without additional training for a new input.

The network has scalability on both the number of the decap candidates \(n\) and that of the decap assignments \(m\). \(n\) is also called the size of the action space. The scalability is due to the context embedding and shared weight properties of the encoder and decoder networks. Details on the shared weight property are mentioned in Section II-B. The scalability can reduce the training time. Because the network trained in the smaller scale of the decap \(n/m\) problems can be used to solve the larger scale of problems with the larger \(n\) and \(m\). Most of the training time is consumed in the impedance calculation by cascading the Z-matrices of PDN and decaps. Therefore, the size of \(n\) and \(m\) directly affects the training time.

A. MDP for PDN Decap Design

Fig. 3 shows the detailed explanation of the MDP parameters—state \(s\), action \(a\), and reward \(r\). Fig. 3(a) is the top view of the HBM VDDQ PDN. The PDN environment is configured of a 3-D grid world, which is divided by 375 × 375 \(\mu m^2\) unit decap-sized unit cells (UDUCs). \(s\) is defined as a set of 4-D vectors \(x_s\) of \(P\) and \(X\). Each \(x\) contains \(x_-, y_,\) and \(z-coordinates\) and whether the port is the probing port or not

\[ s = \{P, X\} = \{x_{p1:p4}, x_{1:n}\} \tag{3} \]

where \(i \in \{p1:p4, 1:n\}\) indicates a node index. \(x_{p1:p4}\) are denoted in blue triangles and \(x_{1:n}\) are in black circles in Fig. 3(a).
The subaction at P transfer impedance reduction at Thus, it can be expressed as the permutation of subactions at assignment sequence as depicted in red circles in Fig. 3(a).

The PDN with decaps respectively; \( Z_{11_wDecap} \) are the self-impedances seen at \( s \) at the time step \( a_k \) of \( X \). The unit decap is a \( 375 \times 375 \times 375 \) \( \mu \text{m}^3 \) sized NMOS equivalent series resistance (ESR). Details on the decap model are described in Section III.

The subaction \( a_k \) is defined as assigning a unit decap at a certain position \( x_{ak} \) at the time step \( t \). \( x_{ak} \) is the elements of \( X \). The unit decap is a \( 375 \times 375 \) \( \mu \text{m}^2 \) sized NMOS decap with 1.055-nF capacitance and 0.7-m\( \Omega \) equivalent series resistance (ESR). Details on the decap model are described in Section III.

\( a = \{ a_{l=1}, a_{l=2}, \ldots, a_{l=m} \} = \{ x_{a1}, x_{a2}, \ldots, x_{am} \} \)

(4)

The policy \( \pi_\theta(a|s) \) is defined as a PDF of \( a \) for the given \( s \). Since \( a \) is represented by the permutation of subactions \( a_t \), it can be factorized by the PDFs of subactions \( p_\theta(a_t|s, a_{1:t-1}) \)

\[
\pi_\theta(a|s) = \prod_{t=1}^{m} p_\theta(a_t|s, a_{1:t-1}).
\]

(6)

B. Attention-Based Encoder–Decoder Transformer Network for Policy Approximation

\( \pi_\theta(a|s) \) is approximated by the encoder–decoder transformer network, as depicted in Fig. 4(a). The main mechanism of the network is the attention. By the attention computation, the encoder embeds the input \( s \) into high-dimensional node embeddings \( h \) and the decoder provides \( a \) by sequentially computing the PDF of the next assignment \( p_\theta(a_t) \).

The attention mechanism is a weighted value passing algorithm between nodes to capture their relationships [31]. Every node can have query \( q \), key \( k \), and value \( v \). \( q \) is the object for questioning to the neighbor nodes and \( k \) is the description or characteristic for each node [30]. Therefore, each node can compute its compatibility \( u \) to the neighbor nodes by scaled dot-product of its \( q \) and \( k \) of the neighbor nodes [31, eq. (11)]. The attention weight \( w \) is computed by taking softmax function to \( u \) [30, eq. (1)], [31, eq. (12)]. Finally, by using \( w \), the next \( h \) can be embedded as the weighted sum of \( v \) of every node [31, eq. (13)]. Therefore, relationships between nodes can be captured in \( h \). This computation is called a scaled dot-product attention or single head attention (SHA). A multihead attention (MHA) is a parallel computation of scaled dot-product attention to learn diverse representation [30]. With \( M \) sets of linear projected \( q, k, \) and \( v \), each head is calculated through SHA. Finally, the concatenated \( M \) number of heads is linearly projected to the final node embedding \( h \).

The main purpose of the encoder transformer network is to embed relationships between the four probing ports \( x_{p4} \) and the \( n \) decap candidates \( x_{RN} \) into \( h \). It converts the 4-D input \( s (d_l) = 4) \) into 128-D \( h (d_l = 128) \). Fig. 4(b) shows the details of the encoder transformer network. Initial node embedding \( h_0 \) is embedded by the linear projection of input node \( x \),

\[
h_0 = W_s x_i + b_s
\]

(7)

where \( i \in \{p1:p4, 1:n\} \) indicates the node index; \( W_s \in \mathbb{R}^{d_l \times d_s} \) and \( b_s \in \mathbb{R}^{d_l} \) are the learnable parameters. The node embeddings \( h_l^i \) of the layer \( l \) are updated sequentially by attention layers. The encoder is configured of \( L \) number of the attention layers. Each of them consisted of one MHA sublayer and one feed-forward (FF) sublayer with residual connections [31]

\[
h_l^i = h_{l-1}^i + \text{MHA}(h_{l-1}^1, h_{l-1}^2, h_{l-1}^3, h_{l-1}^4, h_2^i, \ldots, h_n^i)
\]

(8a)

\[
h_l^{i+1} = h_l^i + \text{FF}(h_l^i)
\]

(8b)

where \( i \in \{1:L\} \) indicates the layer index; \( h_l^i \) is the output of an MHA sublayer; \( h_l^{i+1} \) is the output of an FF sublayer. The residual connection means \( x + f(x) \) for preventing the gradient vanishing [32]. The FF sublayer has one hidden layer with the dimension \( d_r \); hence, there are four learnable parameters \( W_{ff,1}, b_{ff,1}, W_{ff,2}, b_{ff,2} \). An ReLu activation is used for the
where $k \in \{1:M\}$ indicates a head index. $M$ is the number of heads. $d_h$ and $d_o$ are the dimensions of key and value. $W_{Q}^{l, j} \in \mathbb{R}^{d_h \times d_o}$, $W_{K}^{l, j} \in \mathbb{R}^{d_h \times d_o}$, and $W_{V}^{l, j} \in \mathbb{R}^{d_h \times d_o}$ are the learnable parameters to make each head $h_{p1}^{l, k}$, hence $W_{Q}^{l}$, $W_{K}^{l}$, $W_{V}^{l}$ are in $\mathbb{R}^{M \times d_h \times d_o}$. After $q^l$, $k^l$, and $v^l$ are computed, attention weights $a_{p1, j}^{l, k}$ between the first probing node and other nodes are calculated in parallel

$$a_{p1, j}^{l, k} = \text{softmax} \left( \frac{q_{p1, j}^{l, k}^T k_{j}^{l, k}}{\sqrt{d_k}} \right)$$

where $j \in \{p1:p4, 1:n\}$ indicates the node index. Then, each head $h_{p1}^{l, k}$ is computed by the weighted sum of the values by the attention weight

$$h_{p1}^{l, \cdot} = \sum_j a_{p1, j}^{l, k} v_j^{l}$$

where $j \in \{p1:p4, 1:n\}$ indicates the node index. Finally, all the heads are concatenated and linearly projected back into $d_h$ dimension by a learnable parameter $W_{O} \in \mathbb{R}^{d_h \times M \times d_o}$. Also, the residual connection is added

$$h_{p1}^{l} = W_{O} \text{Concat}\left(h_{p1}^{l, 1}, h_{p1}^{l, 2}, \ldots, h_{p1}^{l, M}\right) + h_{p1}$$

where Concat is the concatenation operation.

All the learnable parameters, $W_{r}$, $b_{r}$, $W_{Q}$, $W_{K}$, $W_{V}$, $W_{ff, 1}$, $W_{ff, 2}$, and $b_{ff, 1}$, $b_{ff, 2}$ are shared for each node in the same sublayer, as depicted in Fig. 4(c). Therefore, the encoder transformer network is scalable on the number of the decap candidate nodes $n$, because the nodes can be easily added without redefining the learnable parameters.

Fig. 5 shows the details of the decoder transformer network. The decoder consists of the $m$ number of decoding units, which is equal to the size of $a$. Each decoding unit computes $p_{\theta}(a_{t})$ of the next decap assignment and outputs the assignment position $a_{t}$. A decoding unit is configured of one MHA layer and one SHA layer. Unlike the encoder, only context node embeddings $h_{c}$ and $h_{i}$ become queries $q_{c}$ and $q_{i}$. Then, $p_{\theta}(a_{t} | x, a_{t-1})$ is computed by the attention with keys $k$ from the encoder output node embeddings of the decap candidates $h_{1}^{L}$, $h_{2}^{L}$, $h_{3}^{L}$, $h_{n}^{L}$. Especially, the node embedding of the previous decap assignment $h_{\tilde{a}_{t-1}}^{L}$ is included in $h_{i}$ at every time step, as shown in Fig. 5. Therefore, the previous assignment can be considered to determine $a_{t}$. This is how the proposed method can capture the sequential combinatorial relationship between the decap assignments.

Since $h_{c}$ becomes the query and determines the policy for $a_{t}$, it is a key factor in the decoder. Thus, $h_{c}$ is defined as follows:

$$h_{c} = \text{Concat}\left(h_{c}^{L, \cdot}, h_{p1}^{l}, h_{p2}^{l}, h_{p3}^{l}, h_{p4}^{l}, \tilde{h}^{L}\right)$$

where $h_{\tilde{a}_{t-1}}^{L}$ indicates the node embedding of the previous decap assignment. $h_{p1}^{l}, h_{p2}^{l}, h_{p3}^{l}$, and $h_{p4}^{l}$ indicate the node embeddings of four probing ports $P$. $h_{\tilde{a}}^{L}$ is the mean of all the node embeddings. The dimension of the context node $d_{hc}$ equals to $d_h \times 6$. Therefore, not only the previous assignment but also the positions of the probing ports can be considered when determining $a_{t}$. 

Fig. 4. (a) Encoder–decoder transformer network for policy approximation. (b) Encoder transformer network for embedding input state $s$ into high-dimensional node embeddings $h$. (c) Details of the attention layer of the encoder—MHA sublayer with residual connection.

FF sublayer. Especially in the MHA sublayer of the encoder, all the nodes have their own queries, which are denoted in the red lines, as shown in Fig. 4(b). This is for all the nodes to capture their relationships with each other through the attention.

Fig. 4(c) shows the details of the MHA sublayer. It shows how the node embedding of the first probing port $h_{p1}^{l}$ is updated by the MHA—graphical explanation of (8a). First of all, each node embedding generates its $q_{p1}^{l,k}$, $k_{p1}^{l,k}$, and $v_{p1}^{l,k}$ by linear projection with learnable parameters $W_{Q}^{l,k}$, $W_{K}^{l,k}$, and $W_{V}^{l,k}$

$$q_{p1}^{l,k} = W_{Q}^{l,k} h_{p1}^{l}, \quad k_{p1}^{l,k} = W_{K}^{l,k} h_{p1}^{l}, \quad v_{p1}^{l,k} = W_{V}^{l,k} h_{p1}^{l}$$

where $k_{p1}^{l,k} = \text{Concat}(W_{Q}^{l,k} h_{p1}^{l}, W_{K}^{l,k} h_{p1}^{l}, W_{V}^{l,k} h_{p1}^{l})$ (9)
Finally, assignments is masked by making \( \tanh \) clipping \([33]\). Selecting the locations of the previous \( dhc \) sequence Fig. 5. Decoder transformer network providing the decap assignment to the estimated \( p \) the position, where the probability is maximized—selecting \( q \) the criterion for updating the weights \( \theta \) of the network. Thus, the loss function \( \mathcal{L}(\theta|s) \) is defined as the expectation of a cost function \( L(a) \), which is defined as the negative reward

\[
\mathcal{L}(\theta|s) = \mathbb{E}_{\pi_{\theta}(a|s)}[L(a)] = \mathbb{E}_{\pi_{\theta}(a|s)}[-r(a|s)].
\]

As shown in Fig. 6, the reward \( r(a|s) \) is calculated by the reward estimator, which computes the impedance of the decap-assigned PDN \( Z_{\text{PDN},\text{wDecap}} \). The reward estimator performs cascading the \( Z \)-matrix of the VDDQ PDN \( Z_{\text{PDN}} \) and that of the decap array \( Z_{\text{da}} \) consisting of \( m \) unit decaps. The cascaded ports are the ports \( p \) and \( q \) in \( Z_{\text{PDN}} \) and \( Z_{\text{da}} \), respectively. Then, \( Z_{\text{PDN},\text{wDecap}} \) is calculated using boundary conditions on the ports \( p \) and \( q \) \([35, \text{eqs. (1)-(3)}]\).

Based on the defined \( L(\theta|s) \), the gradient of the loss function \( \nabla_{\theta} \mathcal{L}(\theta|s) \) can be derived as the following equation \([29], [34]\):

\[
\nabla_{\theta} \mathcal{L}(\theta|s) = \mathbb{E}_{\pi_{\theta}(a|s)}[(L(a) - L(a^{\text{BL}}))] \nabla_{\theta} \log \pi_{\theta}(a|s)
\]

where \( L(a^{\text{BL}}) \) is the cost function estimated by the rollout baseline \( \theta^{\text{BL}} \) \([31]\). The baseline network has the same configuration as the transformer policy network. It is periodically updated after every epoch if the improvement is critical according to a paired \( t \)-test \([31]\). The reason why subtracting the baseline is to reduce variance when training \([29]\). Finally, the learnable parameter \( \theta \) of the policy network is optimized by a gradient descent Adam optimizer \([36]\).

The policy network is trained with randomly generated datasets. This is to make reusable on the positions of \( P \) and \( X \). Details on the random dataset generator are explained in the following subsection, and an overall pseudo algorithm for training is shown in Algorithm 1.

D. Random Dataset Generator for Reusability

To train the reusable policy network, a random dataset generator is implemented. It generates the dataset \( s \) and
Algorithm 1 Training via REINFORCE With Rollout Baseline
(Decap n/m)

Inputs: action space size $n$, number of epochs $E$, steps per epoch $T$, batch size $B$, validation size $V$, full-ports PDN Z-matrix $Z_{fp}$, m-sized decap array Z-matrix $Z_{da}$

Initialize $\theta, \theta^{BL}$.

for epoch = 1 to $E$ do
  for step = 1 to $T$ do
    $s_i, Z_{si} \leftarrow \text{RandomDataGen}(n, Z_{fp}) \forall i \in \{1, \ldots, B\}$
    $a_i \leftarrow \text{SampleInference}(s_i, \pi_{\theta}) \forall i \in \{1, \ldots, B\}$
    $a^{BL}_i \leftarrow \text{GreedyInference}(s_i, \pi_{\theta}) \forall i \in \{1, \ldots, B\}$
    $r_i \leftarrow \text{RewardEst}(a^*, s_i, Z_{si}, Z_{da}) \forall i \in \{1, \ldots, B\}$
    $\theta^{BL} \leftarrow \text{GreedyInference}(a^{BL}, s_i, Z_{si}, Z_{da}) \forall i \in \{1, \ldots, B\}$
    $\nabla_{\theta} \mathcal{L} \leftarrow \sum_{i=1}^{B} (L(a_i) - L(a_{i}^{BL})) \nabla_{\theta} \log(\pi_{\theta}(a_i))$
    $\theta \leftarrow \text{Adam}(\theta, \nabla_{\theta} \mathcal{L})$
  end for
  $s_j, Z_{sj} \leftarrow \text{RandomDataGen}(n, Z_{fp}) \forall j \in \{1, \ldots, V\}$
  $a_j \leftarrow \text{GreedyInference}(s_j, \pi_{\theta}) \forall j \in \{1, \ldots, V\}$
  $a^{BL}_j \leftarrow \text{GreedyInference}(s_j, \pi_{\theta}^{BL}) \forall j \in \{1, \ldots, V\}$
  if PairedTTest($\pi_{\theta}(a), \pi_{\theta}^{BL}(a^{BL}) < 0.05$ then
    $\theta^{BL} \leftarrow \theta$
  end if
end for

Corresponding PDN Z-matrices $Z_{PDN}$, $s$ is the set of state $s$ consisting of randomly selected four probing ports $P$ and $n$ decap candidate ports $X$, as shown in Fig. 3(a)

$$Z_{PDN,s} = \{Z_{s1}, Z_{s2}, \ldots, Z_{sN}\}$$ (19)

where $N$ is the number of the data; $k$ indicates the data index; $P_k$ and $X_k$ are defined in (2) and (1), respectively. Each PDN matrix $Z_{sk}$ is generated from a full-port Z-parameter model $Z_{fp}$ of the HBM VDDQ PDN. $Z_{fp}$ has a total of 816 ports on the on-chip and on-interposer PDN since one port is assigned for one UDUC in Fig. 3(a); 48 probing port candidates are in the I/O region (PHY) and 768 decap candidates are in the remaining on-chip and on-interposer PDN. $N = 1000, 100$, and 100 are used for training, validation, and test sets, respectively.

III. MODELING OF HBM VDDQ PDN FOR RANDOM DATASET GENERATOR AND REWARD ESTIMATOR

For implementing the random dataset generator and reward estimator, a VDDQ PDN model based on the HBM gen 2 and 2E is constructed, as shown in Fig. 7(a) [37], [38], [39]. The modeling components considered in this work are as follows: on-chip grid P/G planes, P/G bump array with contact vias, on-interposer meshed P/G planes, multiarray P/G TSVs, package (PKG) PDN, and NMOS decaps. Details on physical dimensions and material properties are summarized in Table I. Because the proposed method is to optimize the on-interposer and on-chip decaps, the precise and distributed models are implemented for on-chip and on-interposer PDNs. However, the lumped circuit model is enough for the PKG PDN to model the hierarchical PDN impedance profile. Also, it is more efficient in terms of computing cost when modeling for this work. The whole hierarchical PDN is modeled by cascading all the modeled components, as shown in Fig. 7(b). The principle of cascading two Z-matrices is the same as described in Section II-C. All the distributed ports of the on-chip and on-interposer components are cascaded. Since PKG PDN is modeled as the lumped model, all the bottom ports of the multiarray TSV model are merged and cascaded to the PKG PDN [13].

The on-chip and on-interposer P/G planes are modeled by cascading the UDCs. The UDCs are modeled by cascading the UCs. Since the size of the UDC is the same for both the chip and interposer, the on-chip UDC is composed of 5625 (75 by 75) on-chip UCs and the on-interposer UDC is configured of 625 (25 by 25) on-interposer UCs. Through W-element modeling methods, the UCs are modeled in per-UC resistance $R_{UC}$, inductance $L_{UC}$, conductance $G_{UC}$, and capacitance $C_{UC}$, as shown in Fig. 8(a) [40], [41]. Both $R_{UC}$ and $G_{UC}$ include dc and ac components ($R_0, R_1, G_0$, and $G_1$). First, $y$-direction wave propagation of the UC is modeled. Using a perfect magnetic conductor boundary on the $x$-direction sides of the UC, a channel-like P/G plane with two ports on the other $y$-direction sides is made [40]. The channel-like P/G...
plane is simulated by 3-D EM simulator to obtain \( S \)-parameter. By assuming it as a W-element transmission line (TL) model, per-length model parameters \( R_0 \), \( R_f \), \( G_0 \), \( G_f \), and \( C \) are extracted from the simulated \( S \)-parameter [41].

When extracting frequency-independent parameters \( R_0 \), \( R_f \), \( G_0 \), \( G_f \), and \( C \) from real parts of per-length impedance \( Z \) and admittance \( Y \), least-square solutions are derived [41, eq. (6)]. When extracting frequency-dependent parameters \( L \) and \( C \) from imaginary parts of \( Z \) and \( Y \), respectively, the mean of the solutions is derived [41, eq. (9)]. The final UC model is obtained by extending the \( x \)-direction model parameters as the same as the \( y \)-direction.

The \( \mu \) bump array in the HBM VDDQ PDN consists of 432-\( \mu \) bump pairs [37]. The coupling between \( \mu \) bump pairs is assumed to be negligible. Then, the model of \( \mu \) bump array \( Z_{\mu \text{ bump Array}}(f) \) is expressed by the extension of that of 1-pair \( \mu \) bumps \( Z_{1-\text{pair}}(f) \)

\[
Z_{\mu \text{ bump Array}}(f) = \begin{bmatrix}
Z_{1-\text{pair}}(f) & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & Z_{1-\text{pair}}(f)
\end{bmatrix}
\]  

(20)

where all the diagonal elements are the two-port \( Z_{1-\text{pair}}(f) \) and zero-matrices \( 0 \) otherwise. As shown in Fig. 8(b), the 1-pair \( \mu \) bumps with contact \( \mu \) vias and (c) NMOS decap.

The unit NMOS decap for the subaction \( a_i \) is modeled by scaling the model of one NMOS decap. Generally, the NMOS decap is modeled as \( C_{\text{decap}} \) and \( ESR_{\text{decap}} \) in series, as shown in Fig. 8(c) [42]. The model is based on the TSMC 65-nm process, and the parameters \( L_{\text{decap}} \) and \( W_{\text{decap}} \) are 0.24 and 1 \( \mu \)m, respectively. \( C_{\text{decap}} \) and \( ESR_{\text{decap}} \) for the

---

**TABLE I**

**Physical Dimensions and Material Properties of HBM VDDQ PDN**

| Objective          | Parameter          | Description                  | Value      |
|--------------------|--------------------|------------------------------|------------|
| On-chip P/G plane  | \( L_{\text{UC, chip}} \) | Length of on-chip UC         | 5 (\( \mu \)m) |
|                    | \( W_{\text{UC, chip}} \) | Width of on-chip UC          | 2 (\( \mu \)m) |
|                    | \( S_{\text{UC, chip}} \) | Space of on-chip UC          | 3 (\( \mu \)m) |
|                    | \( t_{\text{Cu, chip}} \) | Metal (Copper) thickness     | 0.5 (\( \mu \)m) |
|                    | \( h_{\text{Si, chip}} \) | Height of silicon substrate   | 30 (\( \mu \)m) |
|                    | \( h_{\text{IMD, chip}} \) | Height of IMD layer          | 0.5 (\( \mu \)m) |
|                    | \( h_{\text{UBM, chip}} \) | Height of UBM layer          | 2 (\( \mu \)m) |
|                    | \( h_{\text{int, chip}} \) | Height of interposer UC      | 1 (\( \mu \)m) |
| On-interposer P/G  | \( L_{\text{int, interposer}} \) | Length of interposer UC      | 15 (\( \mu \)m) |
|                    | \( W_{\text{int, interposer}} \) | Width of interposer UC       | 7.5 (\( \mu \)m) |
|                    | \( S_{\text{int, interposer}} \) | Space of interposer UC       | 7.5 (\( \mu \)m) |
|                    | \( t_{\text{Cu, interposer}} \) | Metal (Copper) thickness     | 1 (\( \mu \)m) |
|                    | \( h_{\text{Si, interposer}} \) | Height of silicon substrate   | 100 (\( \mu \)m) |
|                    | \( h_{\text{IMD, interposer}} \) | Height of IMD layer          | 1 (\( \mu \)m) |
|                    | \( h_{\text{UBM, interposer}} \) | Height of UBM layer          | 1 (\( \mu \)m) |
| \( \mu \) bump array | \( d_{\text{bump}} \) | Diameter of \( \mu \) bump  | 25 (\( \mu \)m) |
|                    | \( h_{\text{bump}} \) | Height of \( \mu \) bump     | 33 (\( \mu \)m) |
| Contact            | \( d_{\text{via}} \) | Diameter of \( \mu \) via    | 2 (\( \mu \)m) |
| \( \mu \) via      | \( h_{\text{via}} \) | Height of \( \mu \) via       | 1.5 (\( \mu \)m) |
| Multi-array P/G     | \( d_{\text{TSV}} \) | Diameter of TSV              | 15 (\( \mu \)m) |
| TSV                | \( f_{\text{SOI}} \) | SiO\(_2\) thickness          | 0.5 (\( \mu \)m) |
| \( \sigma_{\text{Cu}} \) | | Conductivity of Cu (P/G metal, \( \mu \) bump, contact \( \mu \) via) | 5.8x10\(^7\) (S/m) |
| \( \sigma_{\text{Si}} \) | | Conductivity of Si substrate | 10 (S/m) |
| \( \epsilon_{\text{Cu, chip}} \) | | Relative permittivity of on-chip IMD layer | 3.5 @ 9.0 GHz |
| \( \epsilon_{\text{Si, chip}} \) | | Relative permittivity of on-chip ILD layer | 4.1 @ 9.0 GHz |
| \( \epsilon_{\text{UBM, chip}} \) | | Relative permittivity of on-chip UBM layer | 6.5 @ 9.4 GHz |
| \( \epsilon_{\text{underfill, chip}} \) | | Relative permittivity of on-chip underfill | 3.2 @ 9.4 GHz |
| \( \epsilon_{\text{Si, interposer}} \) | | Relative permittivity of Si interposer | 11.9 @ 9.4 GHz |
| \( \epsilon_{\text{IMD, interposer}} \) | | Relative permittivity of interposer IMD layer | 4.1 @ 9.0 GHz |
| \( \epsilon_{\text{UBM, interposer}} \) | | Relative permittivity of interposer UBM layer | 6.5 @ 9.4 GHz |
| \( \tan\delta_{\text{UBM, chip}} \) | | Loss tangent of on-chip UBM layer | 0.001 @ 9.4 GHz |
| \( \tan\delta_{\text{UBM, interposer}} \) | | Loss tangent of interposer UBM layer | 0.001 @ 9.4 GHz |
| \( \tan\delta_{\text{underfill, chip}} \) | | Loss tangent of on-chip underfill | 0.02 @ 9.4 GHz |
| \( \tan\delta_{\text{IMD, interposer}} \) | | Loss tangent of interposer IMD layer | 0.001 @ 9.4 GHz |

---

Fig. 8. Modeling of (a) on-chip and on-interposer UC, (b) 1-pair P/G \( \mu \) bump with contact \( \mu \) vias, and (c) NMOS decap.
TABLE II
MODELED PARAMETERS OF THE HBM VDDQ PDN

| Objective                        | Parameter       | Value                        |
|----------------------------------|-----------------|------------------------------|
| On-chip P/G plane                | $R_{	ext{onchip}}$, $R_{	ext{onchip}}$ | 30653.63/0.033 (Ω/m)         |
|                                 | $L_{	ext{onchip}}$ | 268.36 (nH/m)                |
|                                 | $G_{	ext{onchip}}$, $G_{	ext{onchip}}$ | 0/1.18×10⁻¹¹ (S/m)           |
|                                 | $C_{	ext{onchip}}$ | 430 (pF/m)                   |
| On-interposer P/G plane          | $R_{	ext{oninterposer}}$, $R_{	ext{oninterposer}}$ | 2699.06/0.033 (Ω/m)          |
|                                 | $L_{	ext{oninterposer}}$ | 120.74 (nH/m)                |
|                                 | $G_{	ext{oninterposer}}$, $G_{	ext{oninterposer}}$ | 0/2.05×10⁻¹¹ (S/m)           |
|                                 | $C_{	ext{oninterposer}}$ | 511.8 (pF/m)                 |
| μ bump array                     | $R_{	ext{μ bump}}$ | 0.04 (Ω @10 GHz)             |
|                                 | $L_{	ext{μ bump}}$ | 8.1 (pH)                     |
| Contact μ via                    | $R_{\text{contact}}$ | 1.2 (Ω)                      |
|                                 | $L_{\text{contact}}$ | 0.75 (pH)                    |
|                                 | $C_{\text{contact}}$ | 0.03 (pF)                    |
| Unit NMOS decap                  | $C_{\text{unit decap}}$ | 1.055 (nF)                   |
|                                 | $\text{ESR}_{\text{unit decap}}$ | 0.7 (mΩ)                     |
| PKG PDN (Lumped)                | $R_{\text{PKG}}$ | 30 (mΩ)                      |
|                                 | $L_{\text{PKG}}$ | 0.5 (nH)                     |
|                                 | $C_{\text{PKG}}$ | 100 (nF)                     |

decap are extracted by SPICE simulator. Then, $C_{\text{unit decap}}$ and $\text{ESR}_{\text{unit decap}}$ are scaled by the number of the decaps in the $375 \times 375 \mu$m² unit area. All the modeled parameters are summarized in Table II.

IV. VERIFICATION OF THE PROPOSED METHOD

In this section, the superiorities of the proposed method are verified by comparing the optimality performance, reusability, scalability, computing time, and cost to the previous methods, including GA, RS, and RL-based methods [21], [22], [23], [24], [25], [26], [27], [28].

Table III summarizes the hyperparameter setup for the transformer network and training. The number of the attention layers in the encoder $L$ is set to 3. The dimensions of the input, embed (hidden), and feedforward layers in the encoder $d_{x}$, $d_{h}$, and $d_{ff}$ are set to 4, 128, and 512, respectively. The dimensions of the context node and hidden layer $d_{hc}$ and $d_{h}$ in the decoder are set to 640 and 128, respectively. The value of tanh clipping $C$ is set to 10. The dimensions of the head, key, and value $M$, $d_{k}$, and $d_{v}$ are set to 8, 16, and 16 for the MHA layers, respectively, and 1, 128, and 128 for the SHA layers, respectively. The total number of 20 train epochs are trained, one epoch size is 1000, the batch size $B$ is 100, and the validation size $V$ is 100. The initial learning rate is $10^{-4}$ and is linearly decayed by $0.95 \times$ every epoch. The proposed method is verified in 12 decap $n/m$ problems with the same hyperparameter setup: decap 100/20, 100/30, 100/50, 100/60, 200/20, 200/40, 200/60, 200/100, 300/20, 300/60, 300/90, and 300/150.

A. Training and Validation Loss Convergence Verification

Fig. 9(a) shows the training and validation loss convergence characteristics in 12 decap $n/m$ problems. For every problem, training and validation loss converge after about 100 training steps (=100 batches = 10 epochs = 10 000 train data).

Fig. 9(b) and (c) shows the performance improvement depending on the training steps in the decap 200/40—the decap optimization results and corresponding self- and transfer impedances of the test data #33. The blue triangles are four probing ports $P$, the black circles are the decap candidates $X$, and the red circles are the decap assignment $a$ in Fig. 9(b). As the training progresses, the policy network assigns unit decaps near the probing ports in both the on-chip and on-interposer PDNs. This result coincides with the physical insight. Corresponding one self-impedance $Z_{22}$ and one transfer impedance $Z_{23}$ are plotted in Fig. 9(c). The probing ports are numbered starting from the lower to the upper port in the PHY region. As the training progresses, both the self- and the transfer impedances are reduced furthermore. From 40- to 100-MHz frequency range, differences in the impedance profiles are not noticeable since the capacitance of the assigned decaps dominates. In other words, the number of the assigned unit decaps is the dominant factor, however spreading loop resistance and inductance dominate in the frequency range over 100 MHz. Those are determined by the positions of assigned unit decaps. Therefore, the results of training step #100 where more unit decaps are assigned closer
to the probing ports and between them show more reduced impedance profiles.

**B. Optimality Performance Verification of the Proposed Method by Comparison to the Conventional GA and RS**

In this subsection, the optimality performance of the proposed method is compared to the conventional GA and RS [16], [28]. The average rewards of the 100 test datasets are compared in all the 12 decap problems. For the GA, the number of the population $P$ and the generation $G$ are 100 and 10, respectively. Moreover, even with 1 sampling width, the average rewards of the proposed method are higher than those of the GA and RS with the sampling width of 1000 in all the problems.

Detailed results of the test data #33 in the decap 200/40 are shown in Fig. 10. The results when all the methods have the sampling width of 1000 are compared—the same computing time for the optimization. Fig. 10(a) depicts the decap optimization results. The distribution of the assigned unit decaps is more confined near the probing ports for both the on-chip and on-interposer PDN in the proposed method than the GA and RS. Fig. 10(b) shows the self- and transfer impedances at the probing port 1 of the initial and optimized PDN by the proposed method. The reduction of $Z_{11}$ indicates the decrement of the self-noise. The reduction

---

**Table IV**

| Method | Proposed Method | GA | RS |
|--------|------------------|----|----|
| Problem | Decap num (m) | Decap num (m) | Decap num (m) |
| **Action space (n)** | **1** | **1000** | **1000** |
| **Sampling width** | | | |
| **100** | 20 | 16.780 | *16.797 | 16.751 | 16.731 |
| | 30 | 18.566 | *18.370 | 18.339 | 18.326 |
| | 50 | 19.929 | *19.934 | 19.917 | 19.910 |
| | 60 | 20.381 | *20.386 | 20.361 | 20.358 |
| **200** | 20 | 16.849 | *16.888 | 16.808 | 16.765 |
| | 40 | 19.438 | *19.448 | 19.354 | 19.329 |
| | 60 | 20.509 | *20.515 | 20.443 | 20.422 |
| | 100 | 21.459 | *21.462 | 21.420 | 21.410 |
| | 20 | 16.874 | *16.928 | 16.836 | 16.774 |
| | 60 | 20.589 | *20.604 | 20.473 | 20.440 |
| | 90 | 21.384 | *21.390 | 21.267 | 21.288 |
| | 150 | 22.055 | *22.057 | 21.979 | 21.991 |

*Best optimality performance*
of the summation of transfer impedances $Z_{12} + Z_{13} + Z_{14}$ indicates that of the transferred noises from ports 2–4. For both the self- and transfer impedances, the assigned decaps reduce the anti-resonance between $L_{PKG}$ and $C_{chip} + C_{interposer}$ around 100 MHz. Also, those reduces the resistance and inductance of the on-chip and on-interposer PDNs over 100 MHz. Fig. 10(c) depicts the comparison of the self- and transfer impedances at port 1 between the proposed method, GA, and RS. The red lines are the results of the proposed method. The black solid lines are those of the GA and the black dashed lines are those of the RS. For both the self- and transfer impedances, the proposed method suppresses impedance more than the GA and RS, especially in the frequency range from 100 MHz to 20 GHz, where loop inductance and resistance are dominant.

C. Reusability and Scalability Verification of the Proposed Method

In this subsection, the reusability and scalability of the proposed method are verified. The average rewards in Table IV are the results of the 100 randomly generated test sets that are not correlated with the training and validation sets. The results by the proposed method show superior performance than the GA and RS. Fig. 11 shows the detailed decap optimization results on four different test data #1, #3, #6, and #21 in decap 200/40. Regardless of the positions of the probing ports $P$ and decap candidates $X$, the reusable policy network provides well-designed results without additional training. Therefore, the transformer-based policy network is reusable on the positions of $P$ and $X$.

Fig. 12(a) shows the verification of the encoder scalability on the size of action space $n$. The method is the same as the encoder scalability verification but sweeping the size of $m$ with the fixed $n$. The network trained in decap 100/20 is applied to solve decap 100/30, 50, and 60; that trained in decap 200/40 to solve decap 200/60 and 100; and that trained in decap 300/60 to solve decap 300/100 and 150. It is to verify whether the network trained in problems assigning 20% of the decap candidates is scalable to solve assigning up to 50%–60% of the decap candidates. The results in Fig. 12(b) show the same optimality performance in all the cases. Therefore, the decoder has the scalability on $m$. 

![Fig. 11. Reusability verification on the positions of the probing ports $P$ and decap candidates $X$—decap optimization results of test data #1, #3, #6, and #21 in decap 200/40.](image1)

![Fig. 12. (a) Verification of encoder scalability on the size of action space $n$. (b) Verification of decoder scalability on the number of the decap assignments $m$.](image2)
D. Computing Time and Cost Comparison to the Conventional GA, RS, and Previous RL-Based Methods

In this subsection, the computing time and cost of the proposed method are compared with the previous methods. The optimization time and cost are compared between the proposed method, GA, and RS. The training time and cost are compared between the proposed method and the previous RL-based methods [21], [22], [23], [24], [25], [26], [27], [28].

To compare with the GA and RS, a time-performance analysis is performed, as shown in Fig. 13. The time-performance analysis is widely used to evaluate the performance improvement of the optimization methods depending on the time or sampling width [33]. The average rewards are compared to find out the values of the sampling width to realize the same performance. For every problem, even with the sampling width of 1000, the GA and RS do not reach the same performance of the proposed method with the sampling width of 1. In other words, the GA and RS need to sample 1000× more to achieve the same performance. Therefore, more than 1000× of the computing time and cost are required in the GA and RS. Detailed comparison results of the optimization time and cost to realize the same performance are summarized in Table V.

For the previous RL-based methods, we cannot directly compare the results on the decap nlm problems, because the previous works are limited in the solution space coverage and only considered the self-impedance in one probing port, as shown in Table VI. In addition, the methods in [21], [22], [23], [24], [25], [26], [27] do not have reusability since the value or policy networks are trained in one PDN data. The networks always must be retrained whenever a new different PDN data is given. Therefore, their optimization time includes the training time, causing a significant increment of the total computing time than the reusable proposed method.

Unlike all the previous RL methods, the proposed transformer network has the scalability. Park et al. [21], Shin et al. [22], Zhang et al. [23], Park et al. [24], Zhang et al. [25], Han et al. [26], Park et al. [27], and Kim et al. [28] must train the networks in the same scale of the decap problems that they intend to solve. However, the proposed transformer network can be trained in the smaller scale of the problems and solve the larger scale problems. Table VII shows the reduction of the training time and data cost depending on the scalability of the encoder and decoder. The results are based on the scalability verification in Fig. 12(a) and (b) of Section IV-C. The data cost is the size of the 1000 training and 100 validation data.

The encoder scalability can reduce both the training time and data cost, because the size of the VDDQ PDN Z-matrix
for the training can be significantly reduced. It leads to the decrement of the execution time for cascading the VDDQ PDN and decap array. The reduction ratio of the training time between decap 100/20 and 200/20 is 80.7%. That between decap 100/20 and 300/20 is 91.9%. The reduction ratios of the data cost are 74.1% and 88.6%, respectively. The decoder scalability can also reduce the training time because the size of the decap array Z-matrix is reduced. The reduction ratio of the training time between decap 100/20 and 60, that between decap 200/40 and 100, and that between decap 300/60 and 150 are 42.1%, 54.8%, and 36.6%, respectively.

V. DISCUSSION

The proposed transformer network can be applied to satisfy the target impedance with the minimum layout area. The target impedance can be any type including widely used simple constant or RL type [18]. Also, it can be derived from the PSII specification [10]. Rather training the network to be specialized to satisfy the certain target impedance as [21], [22], [23], [24], [25], [26], and [27], the proposed network is trained to maximize the impedance reduction for the given number of decap assignments \( m \). Also, the network is trained to be scalable on \( m \). Therefore, the scalable network can provide minimized PDN impedance profiles for any given \( m \). Then, by increasing the decap assignments, the network can find out the minimum number to meet the given target impedance. Fig. 14 shows one example of the optimized self-impedance results by the proposed method to satisfy the RL-type target impedance.

VI. CONCLUSION

In this article, for the first time, we propose a transformer network-based RL method for the PDN decap optimization of HBM. The ability to capture sequential relations of the decap assignments, reusability, and scalability realize superiorities on the optimality performance, computing time, and cost, compared to the conventional GA, RS, and the previous RL-based methods. In addition, the proposed method considers both the self- and transfer impedances seen at the multiple ports as optimization metrics. Therefore, we demonstrate the feasibility of the RL-based method in designing practical PDNs such as the HBM VDDQ PDN.

ACKNOWLEDGMENT

The authors would like to acknowledge the technical support from ANSYS Korea.

REFERENCES

[1] J. U. Knickerbocker et al., “3D silicon integration,” in Proc. 58th Electron. Compon. Technol. Conf. (ECTC), May 2008, pp. 538–543.
[2] M. Min and S. Kadivar, “Accelerating innovations in the new era of HPC, 5G and networking with advanced 3D packaging technologies,” in Proc. Int. Wafer Level Packag. Conf. (IWLP), Oct. 2020, pp. 1–6.
[3] B. Koo, S. Lee, K. Chae, and J. Kim, “Industry-leading high bandwidth memory interface solutions for inference/AI,” in Proc. DesignCon, Santa Clara, CA, USA, Jan. 2020, pp. 1–29.
[4] C. Lee et al., “High bandwidth memory (HBM) with TSV technique,” in Proc. Int. SoC Design Conf. (ISOC), Oct. 2016, pp. 181–182.
[5] K. Cho et al., “Design and analysis of power distribution network (PDN) for high bandwidth memory (HBM) interposer in 2.5 D terabyte/s bandwidth graphics module,” in Proc. IEEE 66th Electron. Compon. Technol. Conf. (ECTC), May 2016, pp. 407–412.
[6] D.-C. Yang, J. Xie, M. Swaminathan, X.-C. Wei, and E.-P. Li, “A rigorous model for through-silicon vias with ohmic contact in silicon interposer,” IEEE Trans. Microw. Wireless Compon. Lett., vol. 23, no. 8, pp. 385–387, Aug. 2013.
[7] T. Kim, C. Jo, and S. Moon, “Signal Integrity(Si) aware HBM2e3 interposer design approach considering y-axis offset between logic and HBM die for HPCI/AL/network applications,” in Proc. IEEE 71st Electron. Compon. Technol. Conf. (ECTC), Jun. 2021, pp. 1270–1275.
[8] N. Chatterjee et al., “Architecting an energy-efficient DRAM system for GPUs,” in Proc. IEEE Int. Symp. High Perform. Comput. Archit. (HPCA), Feb. 2017, pp. 73–84.
[9] X. C. Wei, Z. Z. Ou, E.-X. Liu, and E.-P. Li, “Power integrity analysis of TSV based 3-D integrated circuits,” in Proc. Int. Conf. Microw. Millim. Wave Technol. (ICMMT), May 2012, pp. 1–4.
[10] Y. Sun, J. Kim, M. Ouyang, and C. Hwang, “Improved target impedance concept with jitter specification,” IEEE Trans. Electromagn. Compat., vol. 62, no. 4, pp. 1534–1545, Aug. 2020.
[11] T. Shin et al., “Modeling and analysis of system-level power supply noise induced jitter (PSII) for 4 Gbps high bandwidth memory (HBM) I/O interface,” in Proc. IEEE Electr. Design Adv. Packag. Syst. (EDAPS), Dec. 2021, pp. 1–3.
[12] Z. Xu, Z. Wang, Y. Sun, C. Hwang, H. Delingette, and J. Fan, “Jitter-aware economic PDN optimization with a genetic algorithm,” IEEE Trans. Microw. Theory Tech., vol. 69, no. 8, pp. 3715–3725, Aug. 2021.
[13] F. de Paulis et al., “Impact of chip and interposer PDN to eye diagram in high speed channels,” in Proc. IEEE 2nd Workshop Signal Power Integrity (SPI), Brest, France, May 2018, pp. 1–4.
[14] H. T. To, N. Na, A. Wong, H. Ke, A. K. Sharma, and W. H. Moo, “Top-down jitter specification approach for HBM system optimization,” in Proc. DesignCon, Santa Clara, CA, USA, Jan. 2019, pp. 1–15.
[15] M. Popovich, M. Sotman, A. Kolodny, and E. G. Friedman, “Effective radii of on-chip decoupling capacitors,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 16, no. 7, pp. 894–907, Jul. 2008.
[16] S. Piersanti, E. de Paulis, C. Olivieri, and A. Orlandi, “Decoupling capacitors placement for a multichip PDN by a nature-inspired algorithm,” IEEE Trans. Electromagn. Compat., vol. 60, no. 6, pp. 1678–1685, Dec. 2018.

[17] J. Wang, Z. Xu, X. Chu, J. Lu, B. Ravelo, and J. Fan, “Multiport PDN optimization with the Newton–Hessian minimization method,” IEEE Trans. Microw. Theory Techn., vol. 69, no. 4, pp. 2098–2109, Apr. 2021.

[18] K. Kurnia, G. R. Luevano, T. Wang, S. Orzayat, T. Michalka, and J. L. Drewniak, “Fast algorithm for minimizing the number of decap in power distribution networks,” IEEE Trans. Electromagn. Compat., vol. 60, no. 3, pp. 725–732, Jun. 2018.

[19] J. Y. Choi and M. Swaminathan, “Decoupling capacitor placement in power delivery networks using MFEM,” IEEE Trans. Compon., Packag., Manuf. Technol., vol. 1, no. 10, pp. 1651–1661, Oct. 2011.

[20] M. K. Son, J. Y. Choi, H. Kim, K. H. Jejase, and W. D. Becker, “Demystifying machine learning for signal and power integrity problems in packaging,” IEEE Trans. Compon., Packag., Manuf. Technol., vol. 10, no. 8, pp. 1276–1295, Aug. 2020.

[21] H. Park et al., “Reinforcement learning-based optimal on-board decoupling capacitor design method,” in Proc. IEEE 27th Conf. Electr. Perform. Electron. Packag. Syst. (EPEPS), San Jose, CA, USA, Oct. 2018, pp. 213–215.

[22] J. Shin et al., “Reinforcement learning-based decap optimization method for high-performance solid-state drive,” in Proc. IEEE Int. Joint EMCS/EMC Eur. Symp., Jul. 2021, pp. 718–721.

[23] L. Zhang et al., “Decoupling capacitor selection algorithm for PDN based on deep reinforcement learning,” in Proc. IEEE Int. Symp. Electromagn. Compat., Signal Power Integrity, Jul. 2019, pp. 616–620.

[24] H. Park et al., “Deep reinforcement learning-based optimal decoupling capacitor design method for silicon interposer-based 2.5-D/3-D ICs,” IEEE Trans. Compon., Packag., Manuf. Technol., vol. 10, no. 3, pp. 467–478, Mar. 2020.

[25] L. Zhang, W. Huang, J. Jiang, H. Lin, B.-C. Tseng, and C. Hwang, “An enhanced deep reinforcement learning algorithm for decoupling capacitor selection in power distribution network design,” in Proc. IEEE Int. Symp. Electromagn. Compat. Signal Power Intgrity (EMCSI), Jul. 2020, pp. 245–250.

[26] S. Han, O. W. Bhatti, and M. Swaminathan, “Reinforcement learning for the optimization of decoupling capacitors in power delivery networks,” in Proc. IEEE Int. Joint EMCS/EMC Eur. Symp., Jul. 2021, pp. 544–548.

[27] H. Park et al., “Policy gradient reinforcement learning-based optimal decoupling capacitor design method for 2.5-D/3-D ICs using transformer network,” in Proc. IEEE Electr. Design Adv. Packag. Syst. (EDAPS), Dec. 2020, pp. 1–3.

[28] H. Kim et al., “Deep reinforcement learning framework for optimal decoupling capacitor placement on general PDN with an arbitrary probing port,” in Proc. IEEE 30th Conf. Electr. Perform. Electron. Packag. Syst. (EPEPS), Oct. 2021, pp. 1–3.

[29] P. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, Cambridge, MA, USA: MIT Press, 1998.

[30] A. Vaswani et al., “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 5998–6008.

[31] W. Kool, H. Van Hoof, and M. Welling, “Attention, learn to solve routing problems!” in Proc. Int. Conf. Learn. Represent. (ICLR), 2019, pp. 1–25.

[32] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[33] L. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, “Neural combinatorial optimization with reinforcement learning,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2017, pp. 1–15.

[34] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” Mach. Learn., vol. 8, nos. 3–4, pp. 229–256, 1992.

[35] Y. Kim et al., “Segmentation method based modeling and analysis of a glass package power distribution network (PDN),” Nonlinear Theory Appl., vol. 11, no. 2, pp. 170–188, 2020.

[36] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2015, pp. 1–15.

[37] High Bandwidth Memory DRAM (HBM1, HBM2), Standard JESD235D, 2021.

[38] S. Kim et al., “Signal integrity and computing performance analysis of a processing-in-memory of high bandwidth memory (PIM-HBM) scheme,” IEEE Trans. Compon., Packag., Manuf. Technol., vol. 11, no. 11, pp. 1955–1970, Nov. 2021.

[39] K. Son et al., “A novel through mold plate (TMP) for signal and thermal integrity improvement of high bandwidth memory (HBM),” in IEEE MIT-S Int. Microw. Symp. Dig., Dec. 2020, pp. 1–4.

[40] K. Cho et al., “Fast and accurate power distribution network modeling of a silicon interposer for 2.5-D/3-D ICs with multiarray TSVs,” IEEE Trans. Compon., Packag., Manuf. Technol., vol. 9, no. 9, pp. 1835–1846, Sep. 2019.

[41] J. He et al., “Extracting characteristic impedance of a transmission line referenced to a meshed ground plane,” in Proc. IEEE Int. Symp. Electromagn. Compat. (EMC), Jul. 2016, pp. 651–656.

[42] K. Kim et al., “Modeling and analysis of a power distribution network in TSV-based 3-D memory IC including P/G TSVs, on-chip decoupling capacitors, and silicon substrate effects,” IEEE Trans. Compon., Packag., Manuf. Technol., vol. 2, no. 12, pp. 2057–2070, Dec. 2012.

[43] K. Kim et al., “Deep reinforcement learning-based through silicon via (TSV) array design optimization method considering crosstalk,” in Proc. IEEE Electr. Design Adv. Packag. Syst. (EDAPS), Dec. 2020, pp. 1–3.
Haeyeon Kim received the B.S. degree from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2020, where she is currently pursuing the M.S. degree all in electrical engineering. Her research interests include machine learning (ML) application to power distribution network (PDN) design for 2.5-/3-D ICs.

Taein Shin received the B.S. degree in mechanical engineering from the Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea, in 2018, and the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2020, where he is currently pursuing the Ph.D. degree. His current research interests include the modeling, analysis, and optimization of high bandwidth memory (HBM) I/O interface using machine learning for power supply noise-induced jitter reduction.

Keeyoung Son (Graduate Student Member, IEEE) received the B.S. degree in mechanical engineering and the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2019 and 2021, respectively, where he is currently pursuing the Ph.D. degree. His current research interests include the signal integrity (SI) and thermal integrity (TI) co-design and analysis for 2.5-/3-D ICs.

Boogyo Sim (Graduate Student Member, IEEE) received the B.S. and M.S. degrees from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2017 and 2019, respectively, where he is currently pursuing the Ph.D. degree. His current research interests are design, analyze, and modeling of the wireless power transfer systems with high efficiency and low electromagnetic interference.

Subin Kim received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2015, 2017, and 2021, respectively. He has research experience more than six years in the field of power integrity on active interposer-based 2.5/3 D-ICs. He is currently with the Global Technology Center, Samsung Electronics, Suwon, South Korea. His current research interests include active interposer and integrated voltage regulator designs.

Seungtaek Jeong (Member, IEEE) received the B.E. degree (Hons.) in electrical and electronic engineering from the University of Auckland, Auckland, New Zealand, in 2014, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2017 and 2020, respectively. He is currently a Post-Doctoral Researcher with the Electromagnetic Compatibility Laboratory, Missouri University of Science and Technology, Rolla, MO, USA. His current research interests include modeling of electromagnetic interference and SI/PI analysis for high-speed interconnects.

Chulsoon Hwang (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2007, 2009, and 2012, respectively. He joined Samsung Electronics, Suwon, South Korea, as a Senior Engineer, from 2012 to 2015. In July 2015, he joined the Missouri University of Science and Technology (formerly University of Missouri-Rolla), Rolla, MO, USA, where he is currently an Assistant Professor. His research interests include RF design, signal/power integrity in high-speed digital systems, EMI/electromagnetic compatibility (EMC), hardware security, and machine learning.

Dr. Hwang was a recipient of the Asia-Pacific International Symposium on Electromagnetic Compatibility (AP-EMC) Young Scientist Award, the Google Faculty Research Award, and the Missouri S&T’s Faculty Research Award. He was a co-recipient of the IEEE International Symposium on Electromagnetic Compatibility (IEEE EMC) Best Paper Award, the AP-EMC Best Paper Award, and a two-time co-recipient of the DesignCon Best Paper Award.

Joungho Kim (Fellow, IEEE) received the B.S. and M.S. degrees from Seoul National University, Seoul, South Korea, in 1984 and 1986, respectively, and the Ph.D. degree from the University of Michigan, Ann Arbor, MI, USA, in 1993, all in electrical engineering. He joined the Memory Division, Samsung Electronics, Suwon, South Korea, in 1994, where he was involved in gigabit-scale DRAM design. In 1996, he joined the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, where he is currently a Professor with the Department of Electrical Engineering. He is also the Director of the 3-D Integrated Circuit (IC) Research Center supported by SK Hynix, Inc., Icheon-si, South Korea, and the Smart Automotive Electronics Research Center supported by KET, Inc., Incheon, South Korea. He has given more than 219 invited talks and tutorials in academia and related industries. He has authored or coauthored more than 404 technical papers in refereed journals and conference proceedings. He has authored a book entitled Electrical Design of Through-Silicon-Via (Springer, 2014). In particular, his major research interests include chip package printed circuit board (PCB) codesign and co-simulation for signal integrity, power integrity, ground integrity, timing integrity, and radiated emission in 3D-IC, through-silicon via (TSV), and interposer. His current research interests include electromagnetic compatibility (EMC) modeling, design, and measurement methodologies of 3D-IC, TSV, interposer, system-in-package, multilayer PCB, and wireless power transfer (WPT) technology for 3D-IC.