Reducing Parameter Estimation Error of Behavioral Modeling and Digital Predistortion via Transfer Learning for RF Power Amplifiers

Guichen Yang, Chengye Jiang, *Graduate Student Member, IEEE*,
Renlong Han, *Graduate Student Member, IEEE*, Jingchao Tan, and Falin Liu

***Abstract*— Digital predistortion (DPD) has been widely used in linearizing radio frequency (RF) power amplifiers (PAs). However, model coefficients could not always be estimated accurately for a variety of reasons. Several regularization methods have been developed for parameter identification. However, the performance improvement is limited due to the missing information. Fortunately, if parameters from earlier operating conditions are available, they can be employed to enhance the accuracy of DPD in the current state. Despite the fact that many adaptive DPD methods are based on related concepts, they merely use past parameters as initialization for the target task. In this article, we proposed some novel transfer learning-based parameter estimation techniques for PAs operating in time-varying operating configurations. By effectively utilizing the structure knowledge of noncurrent parameters as a priori rather than just initializing them, the estimation error can be significantly decreased. Applying few-sample learning (FSL), for instance, can help to simplify the computational process of parameter extraction, but its robustness is poor. And the experimental results prove that the proposed method is useful for reducing the parameter estimation bias in FSL with negligible extra computational complexity.**

***Index Terms*— Behavioral modeling, digital predistortion (DPD), few-sample learning (FSL), power amplifiers (PA), transfer learning.**

I. Introduction

WITH the progress of modern wireless communication systems, wider signal bandwidths and higher transmission rates impose severe challenges to the efficiency and linearity of radio frequency (RF) power amplifiers (PAs) [1]. To achieve maximum power efficiency, PAs are often operated in saturation power, which may cause in-band distortion as well as out-of-band spectral regrowth to the transmitted signal [2].

In order to combine both high linearity and high efficiency, digital predistortion (DPD), as an efficient linearization technique, has been widely used in the linearization of RF PAs for wireless communication systems. However, in reality, the nonlinear characteristics of the PA are not constant, which may fluctuate due to aging, temperature drift, and time-varying configurations [3], [4], [5], [6], [7]. And the change in nonlinear characteristics necessitates reestimating the model coefficients.

Most of the widely used DPD models today are linear-in-parameters [8], [9], [10], [11]. As a result, model coefficients can be obtained directly by using least squares (LS). However, in order to reduce computational and hardware costs, model coefficients could not always be accurately estimated due to the lack of sufficient relevant information. Such as parameter extraction from a few samples to avoid large computational complexity [12], [13], [14], [15]; learning from band-limited feedback signals for reducing the sampling rate of analog-to-digital converters (ADCs) in the feedback chain [16], [17], [18], [19], [20]; and parameter estimation through output signal captured by low-bit resolution ADCs [21], [22].

Several regularization methods have been proposed for reducing the error of parameter identification, and some dimensionality reduction techniques may serve a similar purpose, such as ridge regression (RR) [23], [24], generalized RR (GRR) with closed-form solution [25], doubly orthogonal matching pursuit (DOMP) [26], principal component analysis (PCA) [27], and extended manifold regularization (ExMR) [19]. Although the aforementioned methods can improve the accuracy of parameter estimation, the performance is still limited since lost data cannot be retrieved as a whole or to their full advantage.

Similarities in the behavior of PA under different operating conditions may be the key to resolving the lack of information in the current situation since the PA behavior in different configurations is highly correlated [5]. Therefore, we argue that the parameter estimation procedure should not rely solely on current data, model coefficients obtained in the past may also be useful. And it becomes a vital challenge to properly use past coefficients to improve the current prediction.

Applying previous parameters is not a new topic [28], but current approaches often employ them during the initialization phase, and then update parameters using the residuals. We refer to this strategy collectively as residual update-based transfer learning (RUTL) for convenience. Although this concept enhances the efficiency of parameter extraction to a

Manuscript received 6 January 2023; revised 23 February 2023, 16 March 2023, and 10 April 2023; accepted 11 April 2023. Date of publication 27 April 2023; date of current version 6 November 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 61471333. *(Corresponding author: Falin Liu.)*

The authors are with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230026, China (e-mail: noesis@mail.ustc.edu.cn; liufl@ustc.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMTT.2023.3267117.

Digital Object Identifier 10.1109/TMTT.2023.3267117

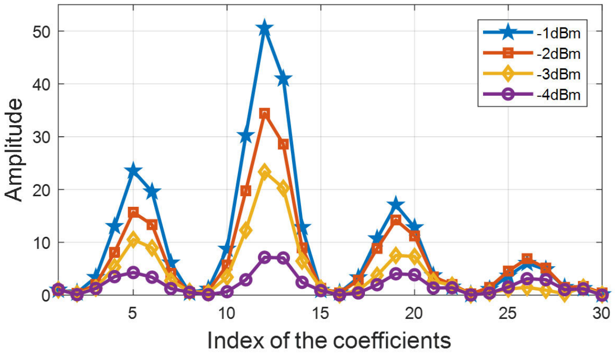


Fig. 1. Amplitude of the forward modeling coefficients for PA in different input power with fixed model structure.

certain amount, it does not fundamentally alleviate the errors present in parameter extraction. In other words, a simple initialization does not utilize past parameters effectively, especially when the training set is biased. In addition, RUTL-based methods are frequently applied to an iterative process, which restricts its application, and the calculation of the residuals requires additional complexity.

The objective of this article is to jump out of typical parameter-based transfer learning and provide an application of previous parameters from a completely different perspective. As the results are shown in Fig. 1, even though the nonlinear behavior of the PA may change (for example, due to the variation in input power), there are commonalities in coefficients when the model order remains unchanged. Put it differently, for the same basic functions, the coefficient weights are close. This implies that the accurate model coefficients produced in the past can be applied to "guide" the present parameter extraction in order to ensure a suitable weight for each basis function, despite the fact that the previous parameters no longer apply to the current circumstance. For example, when few-sample learning (FSL) is applied for parameter identification, due to the statistical mismatch of the training set and the ill-conditioning problem in LS estimation [23], some model parameters may be misestimated. By employing the accurate parameter structure from similar tasks, in this case, it is possible to impose appropriate constraints on each coefficient for more efficient estimation.

In this work, we developed several novel DPD model adaptation techniques based on transfer learning to reduce the parameter extraction error. Instead of estimating model coefficients based on the training set only, three unique strategies are presented to efficiently utilize model coefficients from past tasks (source task), along with a corresponding closed-form solution. By imposing structural prior knowledge in parameter identification of the present task (target task), the estimated coefficients are not significantly biased, particularly when the training set is not ideal. Moreover, the additional computational complexity of the proposed methods is almost negligible. Therefore, there is no effect on real-time systems that demand rapid adaptation of coefficients.

The rest of this article is organized as follows. Section II reviews transfer learning and outlines the benefits that it provides. Section III describes the proposed three transfer learning-based methods. The complexity analysis is available in Section IV. In Section V, several experimental results, including simulation and measurement, are presented. Finally, Section VI summarizes the work in this article.

II. Transfer Learning

Traditional machine learning techniques always rely on training data, i.e., generalization from known data to unknown data. Therefore, the success of learning is strongly dependent on the training data. Unfortunately, when the training set is scarce or inaccurate, the performance of the prediction degrades dramatically. Although numerous regularization strategies have been presented to reduce estimation error, most of these methods focus solely on shrinking the space of solutions based on experience to improve prediction, which does not really address the issue caused by the missing training set.

Missing data for the target task could not be generated out of thin air, yet the information from a distinct but related source domain may be useful. This concept of transfer learning has attracted considerable attention in recent decades. In contrast to the existing feature-based transfer learning method named QR-SVD, which extracts features from source tasks [5], we recommend parameter-based transfer learning since the parameters are frequently simpler to obtain for behavioral modeling and DPD of PA, requiring fewer computational resources consumed in pretraining [29] and sometimes even pretraining free. Besides, the method proposed in [5] prefers the extraction of similar characteristics across diverse conditions in order to lower the computational cost of parameter estimation in a new operation state. However, when the source parameters associated with the target task are insufficient, in other words, the distribution of source tasks is not sufficiently uniform, it is challenging to achieve satisfactory performance in some cases, as the experimental results are shown in Section V.

For convenience, we present the process of transfer learning on forward modeling, which can be considered as the inverse of DPD. Fig. 2 illustrates the flowchart of parameter-based transfer learning, where $\mathbf{X}_S \in \mathbb{C}^{N_S \times P}$ and $\mathbf{X}_T \in \mathbb{C}^{N_T \times P}$ are the basic function matrices of the input signals $\mathbf{x}_S \in \mathbb{C}^{N_S \times 1}$ and $\mathbf{x}_T \in \mathbb{C}^{N_T \times 1}$, respectively. $\mathbf{y}_S \in \mathbb{C}^{N_S \times 1}$ and $\mathbf{y}_T \in \mathbb{C}^{N_T \times 1}$ are the output signals of the PA. N_T and N_S are the number of samples for training, P is the number of coefficients, and the subscripts S and T are used to distinguish the source task from the target task. In addition, $\widehat{\boldsymbol{\beta}}_S$ is the source parameter which can be regarded as a priori knowledge, and $\widehat{\boldsymbol{\beta}}_T$ is the model coefficient required for the target task.

Since the source task can be considered as a linear regression task, the model coefficients can be obtained by LS, which is represented as follows:

$$\widehat{\boldsymbol{\beta}}_S = \left(\mathbf{X}_S^H \mathbf{X}_S\right)^{-1} \mathbf{X}_S^H \mathbf{y}_S \tag{1}$$

where $(\cdot)^H$ is the Hermitian transpose operation.

Although parameters can be calculated in the same way for the target task, LS may not always produce satisfactory results when the training set for the target task is nonideal. Please

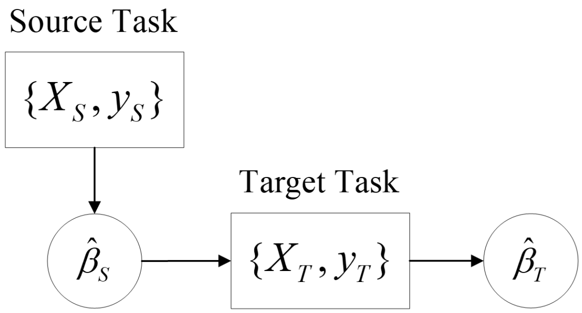


Fig. 2. Procedure of parameter-based transfer learning.

note that we assume the training set of the source task to be ideal, i.e., $\widehat{\boldsymbol{\beta}}_S$ applies to the source task. (It might be possible to obtain the exact parameters using some computationally expensive approaches in the nonideal case, which is beyond the scope of this article). Therefore, the proper utilization of $\widehat{\boldsymbol{\beta}}_S$ to enhance the performance on the target task becomes a crucial issue. In Section III, three effective parameter-based transfer learning methods, including transfer-based generalized ridge regression (TGRR), fine-tuning-based transfer method (FTM), and ridge-type transfer method (RTM), will be introduced to solve this issue. Moreover, their derivation procedure and closed-form solution will be presented.

III. Transfer Learning-Based Parameter Estimation

A. Transfer-Based Generalized Ridge Regression

Before introducing the proposed transfer learning-based GRR, we first describe GRR. The PA model can be expressed as follows:

$$\mathbf{y}_T = \mathbf{X}_T \boldsymbol{\beta}_T + \boldsymbol{\varepsilon} \tag{2}$$

where $\boldsymbol{\varepsilon} \in \mathbb{C}^{N_T \times 1}$ can be regarded as noise. By applying the orthogonal transformation, (2) can be transformed to

$$\mathbf{y}_T = \mathbf{Z}_T \boldsymbol{\alpha}_T + \boldsymbol{\varepsilon} \tag{3}$$

where $\mathbf{Z}_T = \mathbf{X}_T \mathbf{U}$ and $\boldsymbol{\alpha}_T = \mathbf{U}^H \boldsymbol{\beta}_T$. $\mathbf{U} \in \mathbb{C}^{P \times P}$ is an orthogonal matrix, which satisfies $\mathbf{U}^H \mathbf{U} = \mathbf{I}$, and $\mathbf{I} \in \mathbb{R}^{P \times P}$ is the identity matrix. Moreover, $\mathbf{Z}_T$ satisfies

$$\mathbf{Z}_T^H \mathbf{Z}_T = \boldsymbol{\Sigma} = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_P \end{bmatrix} \tag{4}$$

where λ_i is the ith eigenvalue of the Hessian matrix $\mathbf{X}_T^H \mathbf{X}_T$.

The parameter estimated by GRR can be expressed as

$$\widehat{\boldsymbol{\beta}}_T = \mathbf{U}(\boldsymbol{\Sigma} + \mathbf{K})^{-1} \mathbf{Z}_T^H \mathbf{y}_T. \tag{5}$$

Different from RR, $\mathbf{K} \in \mathbb{R}^{P \times P}$ is a diagonal matrix, which shrinks all eigenvalues in $\boldsymbol{\Sigma}$ individually with different penalty parameters, as follows:

$$\mathbf{K} = \begin{bmatrix} k_1 & 0 & \dots & 0 \\ 0 & k_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & k_P \end{bmatrix}. \tag{6}$$

There exists an optimal closed-form solution for $\mathbf{K}$, which can be expressed as

$$k_i = \frac{\sigma^2}{\left\| \alpha_T^{(i)} \right\|^2} \tag{7}$$

where $\|\cdot\|$ is the l_2-norm, and $\alpha_T^{(i)}$ is the ith term of $\boldsymbol{\alpha}_T$. However, although σ^2 can be easily estimated by

$$\widehat{\sigma}^2 = \frac{\left\| \mathbf{y}_T - \mathbf{X}_T \left(\mathbf{X}_T^H \mathbf{X}_T \right)^{-1} \mathbf{X}_T^H \mathbf{y}_T \right\|^2}{N_T - P} \tag{8}$$

the exact $\boldsymbol{\alpha}_T$ could not be available, especially when the sample size for the target task is small. This is the dilemma of the standard GRR, i.e., in order to achieve more precise parameters, the amplitude of each parameter must be determined first. Although some GRR-based derivation methods have been presented [30], [31], [32], their performance remains constrained.

Introducing the concept of "transfer" into GRR could be a feasible strategy for addressing its shortcomings. When the source task is significantly associated with the target task, $\boldsymbol{\beta}_T$ resembles $\widehat{\boldsymbol{\beta}}_S$. This suggests that $\widehat{\boldsymbol{\beta}}_S$ could replace the unknown $\boldsymbol{\beta}_T$ to some extent, and get closer to the optimal solution than alternative approximation solutions. Therefore, (7) can be rewritten as

$$k_i = \frac{\widehat{\sigma}^2}{\left\| \widehat{\alpha}_S^{(i)} \right\|^2} \tag{9}$$

where $\widehat{\alpha}_S = \mathbf{U}^H \widehat{\boldsymbol{\beta}}_S$. The procedures of TGRR are summarized in Algorithm 1.

Algorithm 1 TGRR Algorithm

```
Input: $\mathbf{X}_T \in \mathbb{C}^{N_T \times P}$, $\mathbf{y}_T \in \mathbb{C}^{N_T \times 1}$, $\widehat{\boldsymbol{\beta}}_S \in \mathbb{C}^{P \times 1}$
Output: $\widehat{\boldsymbol{\beta}}_{TGRR} \in \mathbb{C}^{P \times 1}$
Perform orthogonal transformation: $\mathbf{U} = \text{orth}(\mathbf{X}_T^H \mathbf{X}_T)$
$[N_T, P] = \text{size}(\mathbf{X}_T)$
$\boldsymbol{\Sigma} = \mathbf{U}^H \mathbf{X}_T^H \mathbf{X}_T \mathbf{U}$
$\widehat{\sigma}^2 = \left\| \mathbf{y}_T - \mathbf{X}_T \left( \mathbf{X}_T^H \mathbf{X}_T \right)^{-1} \mathbf{X}_T^H \mathbf{y}_T \right\|^2 / (N_T - P)$
$\widehat{\boldsymbol{\alpha}}_S = \mathbf{U}^H \widehat{\boldsymbol{\beta}}_S$
for $i = 1$ to $P$
    $k_i = \widehat{\sigma}^2 / \left\| \alpha_S^{(i)} \right\|^2$
end for
$\mathbf{K} = \text{diag}(k)$
$\widehat{\boldsymbol{\beta}}_{TGRR} = \mathbf{U}(\boldsymbol{\Sigma} + \mathbf{K})^{-1} \mathbf{U}^H \mathbf{X}_T^H \mathbf{y}_T$
```

B. Fine-Tuning-Based Transfer Method

Another easy-to-implement parameter-based transfer method can be considered for fine-tuning the parameters. Model coefficients are initialized to $\widehat{\boldsymbol{\beta}}_S$ and adjusted by applying the batch gradient descent algorithm to solving

$$\min_{\boldsymbol{\beta}_T} \left\| \mathbf{y}_T - \mathbf{X}_T \boldsymbol{\beta}_T \right\|^2. \tag{10}$$

Fine-tuning is an iterative process, initialized as

$$\boldsymbol{\beta}_T^{(0)} = \widehat{\boldsymbol{\beta}}_S. \tag{11}$$

Then, the *iter*-th iteration can be expressed as

$$\boldsymbol{\beta}_T^{(\text{iter})} = \boldsymbol{\beta}_T^{(\text{iter}-1)} - \mu \mathbf{X}_T^H \left(\mathbf{y}_T - \mathbf{X}_T \boldsymbol{\beta}_T^{(\text{iter}-1)}\right) \tag{12}$$

where μ is the learning rate. Assume that the parameters of the target task are estimated by LS

$$\widehat{\boldsymbol{\beta}}_T = \left(\mathbf{X}_T^H \mathbf{X}_T\right)^{-1} \mathbf{X}_T^H \mathbf{y}_T \tag{13}$$

then

$$\mathbf{y}_T \approx \mathbf{X}_T \widehat{\boldsymbol{\beta}}_T. \tag{14}$$

Following (14), (12) can be rewritten as

$$\begin{aligned} \boldsymbol{\beta}_T^{(\text{iter})} &= \boldsymbol{\beta}_T^{(\text{iter}-1)} - \mu \mathbf{X}_T^H \left(\mathbf{X}_T \widehat{\boldsymbol{\beta}}_T - \mathbf{X}_T \boldsymbol{\beta}_T^{(\text{iter}-1)}\right) \\ &= \boldsymbol{\beta}_T^{(\text{iter}-1)} - \mu \mathbf{X}_T^H \mathbf{X}_T \left(\widehat{\boldsymbol{\beta}}_T - \boldsymbol{\beta}_T^{(\text{iter}-1)}\right). \end{aligned} \tag{15}$$

By setting

$$\boldsymbol{\Phi} = \mathbf{I} - \mu \mathbf{X}_T^H \mathbf{X}_T \tag{16}$$

we can express (15) as

$$\boldsymbol{\beta}_T^{(\text{iter})} = \boldsymbol{\Phi} \boldsymbol{\beta}_T^{(\text{iter}-1)} + (\mathbf{I} - \boldsymbol{\Phi}) \widehat{\boldsymbol{\beta}}_T. \tag{17}$$

Finally, by applying the induction hypothesis, the iterative process of batch gradient descent algorithm leads to a one-step analytical solution, as follows:

$$\boldsymbol{\beta}_T^{(\text{iter})} = \boldsymbol{\Phi}^{\text{iter}} \widehat{\boldsymbol{\beta}}_S + \left(\mathbf{I} - \boldsymbol{\Phi}^{\text{iter}}\right) \widehat{\boldsymbol{\beta}}_T. \tag{18}$$

The accuracy of fine-tuning is dependent on the learning rate μ and the number of iterations *iter*, which can be obtained through traversal search. With consistent statistical features of the transmitted signal, our experience indicates that the difference between the optimal hyperparameters is modest, meaning that the search procedure needs to be done only once. Furthermore, a detailed analysis of the hyperparameters is available in Section V-E. Please note that $\boldsymbol{\Phi}$ and $\boldsymbol{\Phi}^{\text{iter}}$ can be pretrained and fixed when μ and *iter* are determined. Therefore, the additional computational cost required for model coefficients adaptation is very low.

The full procedures are detailed in Algorithm 2.

Algorithm 2 FTM Algorithm

```
Input: $\mathbf{X}_T \in \mathbb{C}^{N_T \times P}$, $\mathbf{y}_T \in \mathbb{C}^{N_T \times 1}$, $\widehat{\boldsymbol{\beta}}_S \in \mathbb{C}^{P \times 1}$, $iter$, $\mu$
Output: $\widehat{\boldsymbol{\beta}}_{FTM} \in \mathbb{C}^{P \times 1}$
$\widehat{\boldsymbol{\beta}}_T = \left(\mathbf{X}_T^H \mathbf{X}_T\right)^{-1} \mathbf{X}_T^H \mathbf{y}_T$
$\boldsymbol{\Phi} = \mathbf{I} - \mu \mathbf{X}_T^H \mathbf{X}_T$
$\boldsymbol{\Phi}^{iter} = \underbrace{\boldsymbol{\Phi} \times \boldsymbol{\Phi} \times \boldsymbol{\Phi} \ldots \times \boldsymbol{\Phi}}_{iter}$
$\widehat{\boldsymbol{\beta}}_{FTM} = \boldsymbol{\Phi}^{iter} \widehat{\boldsymbol{\beta}}_S + \left(\mathbf{I} - \boldsymbol{\Phi}^{iter}\right) \widehat{\boldsymbol{\beta}}_T$
```

C. Ridge-Type Transfer Method

The proposed RTM is inspired by RR, therefore with a similar loss function and closed-form solution to RR. The loss function of RR can be expressed as

$$\mathcal{L}_{\text{RR}} = \left\| \mathbf{y}_T - \mathbf{X}_T \boldsymbol{\beta}_{\text{RR}} \right\|^2 + \lambda_{\text{RR}} \left\| \boldsymbol{\beta}_{\text{RR}} \right\|^2 \tag{19}$$

where λ_{RR} is the ridge parameter of RR, which shrinks $\boldsymbol{\beta}_T$ since a large value of model coefficients usually corresponds to a large error in parameter estimation due to the multicollinearity [33]. Therefore, considering the relevance of the target task to the source task, when the target parameters are estimated inaccurately, there might be a large difference compared with $\widehat{\boldsymbol{\beta}}_S$, which satisfies

$$\left\| \widehat{\boldsymbol{\beta}}_T^{(\text{Inaccurate})} - \widehat{\boldsymbol{\beta}}_S \right\|^2 > \left\| \widehat{\boldsymbol{\beta}}_T^{(\text{Accurate})} - \widehat{\boldsymbol{\beta}}_S \right\|^2. \tag{20}$$

This means that appropriately constraining the difference between the source parameters and the target parameters is beneficial. Thus, (19) can be rewritten as

$$\mathcal{L}_{\text{RTM}} = \left\| \mathbf{y}_T - \mathbf{X}_T \boldsymbol{\beta}_{\text{RTM}} \right\|^2 + \lambda_{\text{RTM}} \left\| \boldsymbol{\beta}_{\text{RTM}} - \widehat{\boldsymbol{\beta}}_S \right\|^2 \tag{21}$$

where λ_{RTM} is the penalty parameter of RTM. The optimization problem can be expressed as

$$\widehat{\boldsymbol{\beta}}_{\text{RTM}} = \arg\min_{\boldsymbol{\beta}_{\text{RTM}}} \mathcal{L}_{\text{RTM}}. \tag{22}$$

To solve (22), let

$$\frac{\partial \mathcal{L}_{\text{RTM}}}{\partial \beta_{\text{RTM}}} = 0. \tag{23}$$

Then,

$$2\mathbf{X}_T^H \mathbf{X}_T \boldsymbol{\beta}_{\text{RTM}} - 2\mathbf{X}_T^H \mathbf{y} + \mathbf{2}\lambda_{\text{RTM}} \boldsymbol{\beta}_{\text{RTM}} - 2\lambda_{\text{RTM}} \widehat{\boldsymbol{\beta}}_S = 0. \tag{24}$$

Finally, the analytic solution is available for the minimizer of the loss function in (21), which can also be regarded as a simplified form of the GRR mentioned in [33], as follows:

$$\widehat{\boldsymbol{\beta}}_{\text{RTM}} = \left(\mathbf{X}_T^H \mathbf{X}_T + \lambda_{\text{RTM}} \mathbf{I}\right)^{-1} \left(\mathbf{X}_T^H \mathbf{y}_T + \lambda_{\text{RTM}} \widehat{\boldsymbol{\beta}}_S\right). \tag{25}$$

The optimal penalty parameter λ_{RTM} may also be obtained by traversal, and it is easier to find the optimal solution since only one hyperparameter is required compared to FTM. Moreover, the additional computational cost compared to LS is almost negligible, as just a few additional addition operations are required.

The detailed method is shown in Algorithm 3. Note that an ideal λ_{RTM} is much larger than ridge parameter λ_{RR} since it restricts the 2-norm of the deviation rather than that of the parameters themself.

Algorithm 3 RTM Algorithm

```
**Input:** $\mathbf{X}_T \in \mathbb{C}^{N_T \times P}$, $\mathbf{y}_T \in \mathbb{C}^{N_T \times 1}$, $\widehat{\boldsymbol{\beta}}_S \in \mathbb{C}^{P \times 1}$, $\lambda_{RTM}$
**Output:** $\widehat{\boldsymbol{\beta}}_{RTM} \in \mathbb{C}^{P \times 1}$
1: $\widehat{\boldsymbol{\beta}}_{RTM} = \left(\mathbf{X}_T^H \mathbf{X}_T + \lambda_{RTM}\mathbf{I}\right)^{-1}(\mathbf{X}_T^H \mathbf{y}_T + \lambda_{RTM}\widehat{\boldsymbol{\beta}}_S)$
```

IV. Complexity Analysis

To illustrate the low additional computational complexity of the proposed methods, a detailed complexity analysis is presented in this section.

A. Complexity of Standard LS

When LS is directly used for parameter identification, the computational complexity can be expressed as

$$O_{\text{Add}}^{(\text{LS})} = 14N_T P^2 + 5N_T P + 5P^3 - 2P^2 - 2P \tag{26}$$

$$O_{\text{Mult}}^{(\text{LS})} = 6N_T P^2 + 3N_T P + 3P^3 \tag{27}$$

where O_{Add} means the number of real additions and O_{Mult} represents that of the real multiplication.

B. Complexity of TGRR

Although several operations in TGRR, such as the orthogonal transformation and the estimation of σ^2, are time-consuming when the statistics of the transmission signal do not change, the transformation matrix $\mathbf{U}$ can be fixed, allowing $\mathbf{U}$ to be calculated only once offline. In addition, since σ^2 represents the energy of the noise, which is approximately constant over a long period of time. Therefore, these two components are disregarded in the complexity analysis, and the final computational complexity can be calculated as follows:

$$O_{\text{Add}}^{(\text{TGRR})} = 14N_T P^2 + 5N_T P + 14P^3 + P^2 - P \tag{28}$$

$$O_{\text{Mult}}^{(\text{TGRR})} = 6N_T P^2 + 3N_T P + 6P^3 + 5P^2 + 3P. \tag{29}$$

C. Complexity of FTM

Similarly, once the hyperparameters *iter* and μ have been determined, the fine-tuning matrix $\boldsymbol{\Phi}^{\text{iter}}$ can be calculated offline, and its complexity can be described as

$$O_{\text{Add}}^{(\text{FTM})} = 14N_T P^2 + 5N_T P + 5P^3 + 12P^2 - 4P \tag{30}$$

$$O_{\text{Mult}}^{(\text{FTM})} = 6N_T P^2 + 3N_T P + 3P^3 + 6P^2. \tag{31}$$

D. Complexity of RTM

Since RTM does not require any pre-training when the hyperparameter λ_{RTM} is fixed, it required only $4P$ more additional real additions than LS. The entire number of real-valued additions and multiplications can be expressed as

$$O_{\text{Add}}^{(\text{RTM})} = 14N_T P^2 + 5N_T P + 5P^3 - 2P^2 + 2P \tag{32}$$

$$O_{\text{Mult}}^{(\text{RTM})} = 6N_T P^2 + 3N_T P + 3P^3. \tag{33}$$

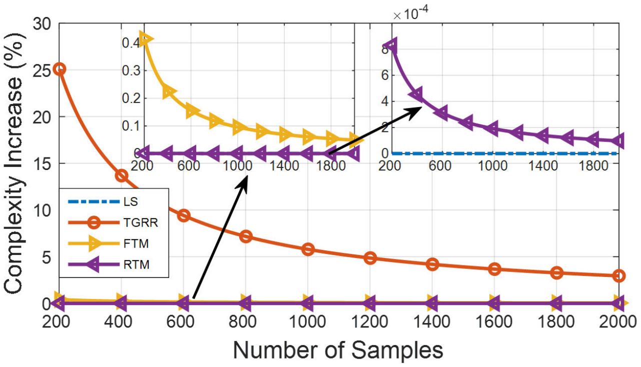


Fig. 3. Computational complexity comparison of the proposed methods with standard LS with the number of parameters $P = 100$.

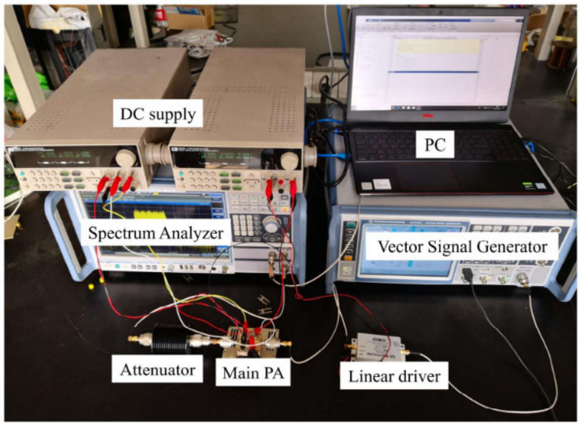


Fig. 4. Photograph of the experimental platform.

E. Complexity Comparison

To effectively assess the computational complexity of the aforementioned methods, the number of floating-point operations (FLOPs) is used as an evaluation metric, with each real addition and real multiplication representing one FLOP [34]. We take the FLOPs required for standard LS as a baseline and calculate the percentage increase of each method compared with LS. The results are shown in Fig. 3, in which P= 100 and N_T is swept from 200 to 2000.

Even after ignoring the computational cost of σ^2 and $\mathbf{U}$, TGRR's complexity remains significant, but it has the advantage of not requiring any hyperparameters. RTM is the least complicated and requires only one hyperparameter adjustment. Moreover, the complexity of FTM falls somewhere in the middle, yet its strength resides in scalability. As seen in (18), FTM does not restrict the acquisition of $\widehat{\boldsymbol{\beta}}_T$. In spite of the fact that LS is used to estimate $\widehat{\boldsymbol{\beta}}_T$ in this article, it can be replaced by a variety of different techniques to further improve the performance and reduce the computational complexity in parameter identification.

V. Experimental Results

A. Experimental Setup

To validate the proposed methods, a test platform was set up, as shown in Fig. 4, which consists of the main PA undertested (BLM9D2325-20AB) operating at 2.4 GHz, a linear

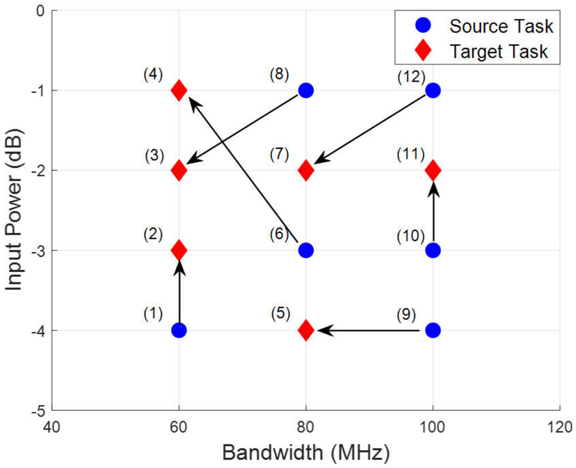


Fig. 5. Illustration of the source task and the target task in Section V-B, where arrows indicate the direction of transfer learning.

TABLE I
INDEX OF THE TIME-VARYING CONFIGURATIONS

Index	Bandwidth	Input Power
1	60 MHz	-4 dBm
2		-3 dBm
3		-2 dBm
4		-1 dBm
5	80 MHz	-4 dBm
6		-3 dBm
7		-2 dBm
8		-1 dBm
9	100MHz	-4 dBm
10		-3 dBm
11		-2 dBm
12		-1 dBm
13		0 dBm

driver amplifier, a vector signal generator (SMW200A), a 40-dB RF attenuator, and a spectrum analyzer (FSW43).

Since it is difficult to accurately analyze changes in the nonlinear characteristics of the PA due to aging or temperature drift, time-varying configurations are used to explore changes in PA behavior. Specifically, we evaluate the performance of the proposed methods in switching between different operating conditions. As a proof of concept, the average input power of the linear driver amplifier varies from -1 to -4 dBm, with a 1-dB step size. The gain of the linear driver amplifier is 10 dB, resulting in a 2 dB maximum gain compression of the PA. The input signal is the orthogonal frequency division multiplexing (OFDM) signal with a 7.2-dB peak-to-average power ratio (PAPR) and bandwidth of 100, 80, and 60 MHz, where 50 000 samples are divided for the training set and another set of 50 000 samples are employed for validation. In addition, the sampling rate of the signal is 368.64 Msamples/s.

The index of time-varying configurations is listed in Table I, including a challenging scenario (state 13) for the following DPD test. In addition, six cases for pretraining (source tasks) and validation (target tasks) are illustrated in Fig. 5, which are divided randomly, and arrows indicate the direction of transfer learning. Note that for a target task, the selection of the source task complies with the following two principles: selecting the most similar source task, and when multiple similar source tasks exist, giving preference to the source task with wider bandwidth or lower input power. For the instance of bandwidth, it is evident that wider bandwidths typically hold more information and are hence downward compatible. In contrast, model coefficients with a high input power always perform a large magnitude, which may increase the risk of overfitting. Section V-C provides a detailed analysis. It is worth noticing that the more challenging 8 → 3 and 6 → 4 are chosen as the transfer direction rather than 6 → 3 and 8 → 4 to better demonstrate the feasibility of the proposed methods.

In the following test, it is assumed that the source task's model coefficients have been accurately estimated and are accessible. Moreover, for the target task, FSL is applied for parameter identification to fully demonstrate the potential of the proposed methods when the training set is scarce. It is worth mentioning that although the proposed transfer learning-based methods increase the computational complexity of FSL and require some additional storage resources, which seem to defeat the original purpose of FSL. These minor resource consumptions are trivial, however, when compared with the previous approaches, which required more samples for training to get similar results.

The model for behavioral modeling and DPD is the generalized memory polynomial (GMP) model [9], which can be expressed as

$$\begin{aligned} y_{\mathrm{GMP}} = &\sum_{k=1}^{K}\sum_{m=0}^{M} a_{k,m}x(n-m)|x(n-m)|^{k-1} \\ &+ \sum_{k=2}^{K}\sum_{m=0}^{M}\sum_{l=1}^{L} b_{k,m,l}x(n-m)|x(n-m-l)|^{k-1} \\ &+ \sum_{k=2}^{K}\sum_{m=0}^{M}\sum_{l=1}^{L} c_{k,m,l}x(n-m)|x(n-m+l)|^{k-1}. \end{aligned} \tag{34}$$

In this work, $M = 5$, $K = 7$, and $L = 2$ are chosen as the model order for all tasks, which yields the number of coefficients $P = 186$. And the architecture for identifying the DPD parameter is indirect learning (IDL) structure [35]. Notably, the effective usage of the source parameters relies on the same model structure, which may result in a slight loss of model accuracy. Because an identical M, K, and L setting for different configurations makes it difficult to ensure that the selected DPD model is optimal for all transmission setups. Fortunately, as the result shown in Fig. 6, the difference between the optimal model and the same model is not so significant. Furthermore, as demonstrated by the following experimental results in Section V-B, the employment of the optimal model does not bring a significant performance gain to the previous methods, and determining the ideal model relies on a large number of traversals which means it is not always available.

During the performance comparison, the normalized mean square error (NMSE) is employed to verify the accuracy of parameter extraction of different methods. Furthermore, while comparing DPD performance, an additional metric of adjacent channel power ratio (ACPR) is also detailed.

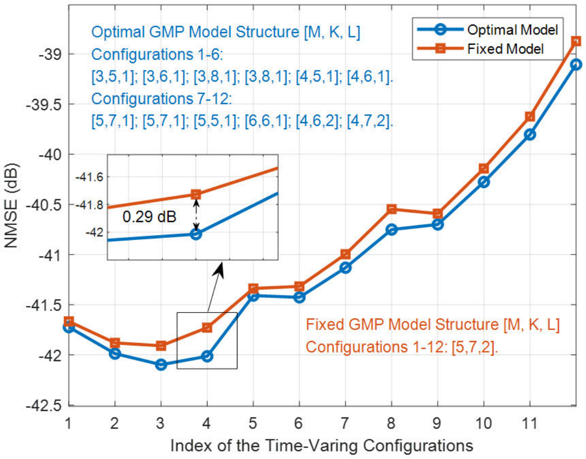


Fig. 6. Comparison of the fixed model and the optimal model in different operation configurations contains detailed model structures.

TABLE II
BEHAVIORAL MODELING RESULTS OF FSL

Method	Source Task → Target Task					
	12→7	10→11	9→5	8→3	6→4	1→2
w/o Update	-29.23	-26.15	-28.53	-24.80	-20.59	-27.04
LS	-36.51 (-36.69)	-35.42 (-36.04)	-37.74 (-38.92)	-37.77 (-38.71)	-38.36 (-38.72)	-37.84 (-39.69)
RR [24]	-36.59 (-36.62)	-35.50 (-35.43)	-39.32 (-39.23)	-37.92 (-38.13)	-38.40 (-38.00)	-39.47 (-39.29)
AGRR [25]	-38.42 (-38.81)	-36.55 (-36.48)	-39.55 (-39.94)	-39.92 (-40.17)	-38.51 (-38.75)	-40.45 (-40.69)
PCA [27]	-38.16 (-38.32)	-36.18 (-36.39)	-39.64 (-40.00)	-39.23 (-39.47)	-38.19 (-38.41)	-40.36 (-40.72)
QR-SVD [5]	**-39.52**	**-37.74**	-40.67	-37.95	-38.21	-38.14
TGRR	-38.66	-36.90	-39.95	-40.34	-39.26	-40.62
FTM	-39.02	-37.26	**-40.97**	-40.31	-39.42	**-40.88**
RTM	-39.05	-37.31	-40.94	**-40.39**	**-39.45**	**-40.88**
Full-Sample Ref.	-41.00 (-41.13)	-39.63 (-39.80)	-41.34 (-41.41)	-41.91 (-42.10)	-41.73 (-42.02)	-41.88 (-41.98)

B. Experimental Results of Behavioral Modeling

In the forward modeling experiments, 300 samples were selected by memory feature-based (MFB) OptiSim [15]. To evaluate the suggested transfer methods, we compared the proposed TGRR, FTM, and RTM to LS, RR [24], PCA [27], alternative GRR (AGRR) [25], and QR-SVD [5]. Since the performance of PCA is heavily dependent on the reduced dimension, we traverse it and present the best result for comparison. For QR-SVD, the pretraining set consists of the six source tasks shown in Fig. 5, and all features are retained to enable a fair comparison, leading to a feature transformation matrix $\mathbf{A} \in \mathbb{C}^{186\times 6}$. In addition, the performance when the full training set is used (Full-Sample Ref.), and the results where the model coefficients from the source task are applied directly to the target task (w/o Update) are also provided as references. The results are detailed in Table II.

In order to better show the benefit of the proposed methods, for approaches that are easy to adjust the model structure (LS, RR, AGRR, PCA, and Full-Sample Ref.), we also provide their performance when employing the optimal model settings in corresponding configurations, and the NMSE results are detailed in parentheses in Table II. It is worth mentioning that since the proposed strategy is based on a one-to-one transfer process, which only requires the same model structure for the source task and the target task rather than using the same model in all configurations. Therefore, the suggested methods seem more flexible than previous QR-SVD.

It can be demonstrated that the proposed methods attain the top in most of the selected transfer scenarios. Note that although QR-SVD produces outstanding results in states 7 and 11, it relies heavily on the dense distribution of source tasks. And its performance is unsatisfactory in the absence of sufficient source parameters around it, as the results in configurations 2, 3, and 4. Besides, QR-SVD has a high cost of pretraining because it requires the QR decomposition of a quite large matrix. In comparison to QR-SVD, the proposed methods need almost no pretraining and provide steady performance.

Notably, the parameter extraction complexity of QR-SVD is significantly lower than that of LS as a result of the shared features, so the computational resource consumption in parameter identification is much less than that of the proposed method. However, this does not imply that the proposed methods have lost their competitiveness. If the high pretraining cost is left aside, QR-SVD should not be considered as a stand-alone method. In other words, the proposed FTM can be used in combination with QR-SVD benefiting from its excellent scalability (fine-tuning the results of QR-SVD). Besides, transformed coefficients in QR-SVD can also be estimated by TGRR and RTM effectively, since the same property as the model coefficients, i.e., similar parameter weights in different configurations. Combining QR-SVD with the proposed methods, a more robust result can be obtained while ensuring a very low parameter estimation complexity.

C. Transfer Learning With Different Source Tasks

For transfer learning-based methods, the success of the transfer is directly determined by the similarity between the source and target tasks. When two domains are too different, transfer learning may not produce significant results and even result in a negative transfer.

To better illustrate this phenomenon, the forward modeling results of the PA using 300 samples for training are provided. As depicted in Fig. 7, the target task is set to the behavioral modeling of the PA at configuration 6 (80 MHz, −3 dBm), and the source parameters are treated to be the ideal model coefficients for all the remaining states. The NMSE results of RTM and "w/o Update" are illustrated in Fig. 8.

The greater the disparity between the source task and the target task, the less the performance gains through transfer learning. Interestingly, the direct application of source parameters on the target task (w/o Update) is also in line with this trend, as indicated by the NMSE results shown in Fig. 8(b). In other words, when the performance of "w/o Update" is poor, meaning a substantial gap emerges between the source task and the target task, which may impact the result of transfer learning since it is difficult to obtain sufficient information from the source task that is applicable for the target task. Therefore, the performance of "w/o Update" can be considered

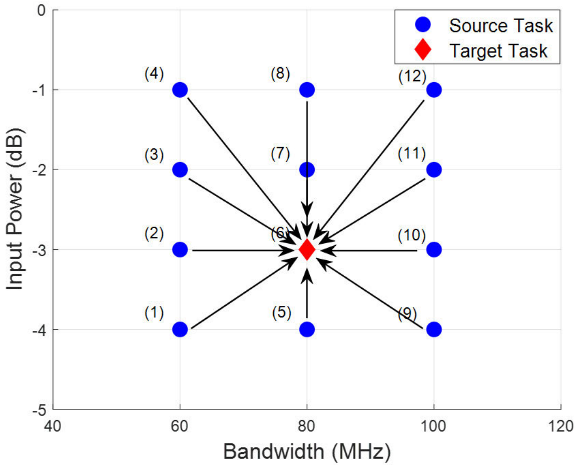


Fig. 7. Illustration of the source task and the target task in Section V-C, where arrows indicate the direction of transfer learning.

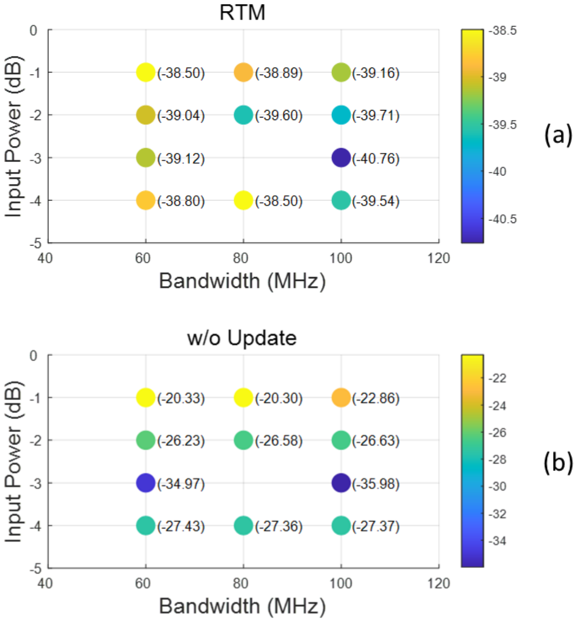


Fig. 8. (a) NMSE results of the target task by applying RTM and (b) directly applying model coefficients from source tasks in Section V-C.

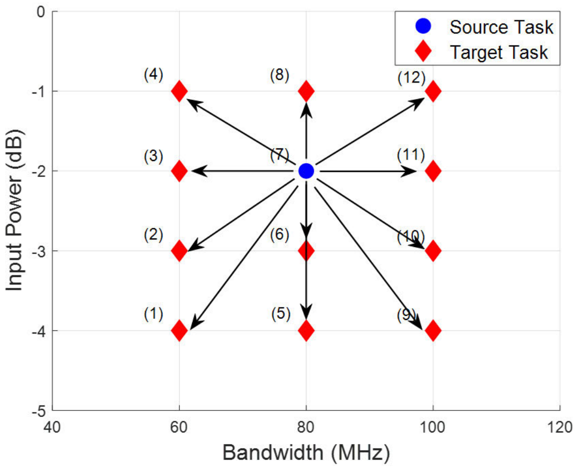


Fig. 9. Illustration of the source task and the target task in Section V-D, where arrows indicate the direction of transfer learning.

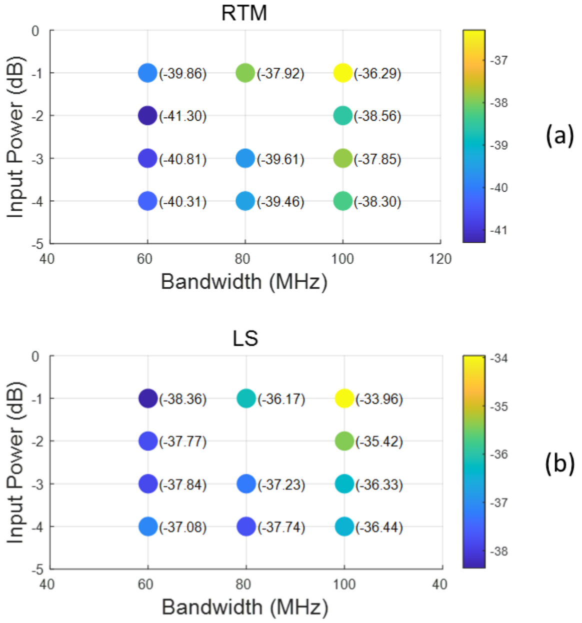


Fig. 10. (a) NMSE results of the target task by applying RTM and (b) applying regular LS in Section V-D.

as a good indicator to evaluate whether the source parameters are sufficiently reliable. Based on our experience, the negative transfer is largely unlikely to occur when the NMSE metric for "w/o Update" is below −20 dB.

In this article, time-varying configurations are the cases mainly considered, whereas PA characteristics for a fixed transmission configuration change much more slowly due to temperature, aging, and so on. In other words, when the operating configuration is fixed, we can assume a tolerable NMSE baseline of −32 dB, DPD model coefficients should be updated when the NMSE performance is close to the baseline rather than worse than it so as to ensure reliable transmission of information. This NMSE below the baseline indicates that the source task (last running DPD coefficients in this case) and the target task are closely similar (performance of "w/o Update" lower than −32 dB), which is better for the application of the proposed methods. However, there are still drawbacks with this application in continuously updating since the source parameters may accumulate errors due to the noise or nonoptimal hyperparameters, we leave it as future work.

D. Transfer Learning With Only One Source Task

In this section, we reduce the number of source tasks to one to better exploit the potential of the proposed method. During this test, 300 samples are selected for parameter identification, the source parameter is the forward modeling coefficients of the PA at state 7, and the target tasks are set to the all-remaining configurations, as shown in Fig. 9. The NMSE results are shown in Fig. 10, in which the accuracy of LS is set as a baseline to adequately compare with the proposed RTM.

Clearly, since the proposed methods are based on one-to-one transfer learning, they do not require as many pretraining tasks as the previous QR-SVD to get satisfactory results.

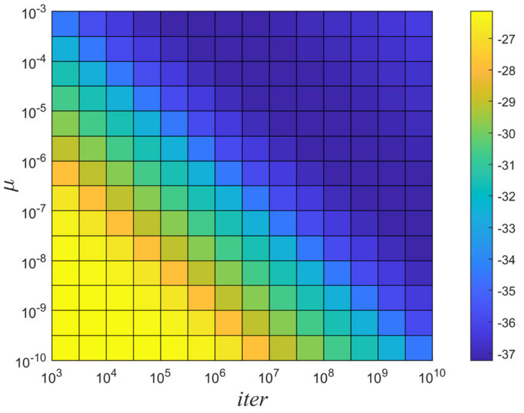


Fig. 11. NMSE results of FTM with different hyperparameters: *iter* ranges from 10^3 to 10^{10} and μ ranges from 10^{-10} to 10^{-3}.

In addition, the suggested approaches are easy to adapt in practical development. Specifically, even if the PA characteristics may vary for reasons other than transmission configuration, such as temperature, compared with the previous QR-SVD which requires a large amount of computational and storage resources to update the transformation matrix, the suggested method just involves reextracting source parameters, and the required computational complexity is relatively much lower.

E. Discussion of Hyperparameters in FTM and RTM

Although the proposed RTM and FTM exhibit good performance, it still depends on a prudent decision of appropriate hyperparameters. In this section, we will discuss on hyperparameters to show how it affect the results of FTM and RTM.

1) Hyperparameters in FTM: For FTM, the hyperparameters are the learning rate μ and the number of iterations *iter* in the fine-tuning process. Since the proposed FTM is fundamentally based on gradient descent, these two hyperparameters are complementary. Simply put, when the learning rate decreases, more iterations are required; while the number of necessary iterations then reduces when the learning rate increases.

To better explain this concept, some behavioral modeling experiments are performed in which model parameters are transferred from state 10 to state 11, with 300 samples for training. In this test, $\mu = 10^{-10}, 10^{-9.5}, \ldots, 10^{-3}$, and $iter = 10^3, 10^{3.5}, \ldots, 10^{10}$. The NMSE results are shown in Fig. 11. Clearly, when $\mu \times iter \approx 10^3$, FTM reaches near-optimal performance. This conclusion applies, in our experience, not just to the current case ($10 \rightarrow 11$), but also to the other scenarios introduced in this article.

2) Hyperparameter in RTM: Since there is a similarity between RTM and RR, the extensively employed ridge trace method in RR is also helpful for analyzing the hyperparameter λ_{RTM} in RTM. We examined three scenarios, including behavioral modeling of PA based on transfer learning from state 10 to state 11, state 12 to 7, and state 6 to 4, with the number of training samples ranging from 200 to 400. In addition, the range of $\lambda_{\text{RTM}} = 2^1, 2^0, 2^{-1}, \ldots, 2^{-20}$. Forward modeling results of the proposed RTM under different situations and hyperparameters are provided in Fig. 12.

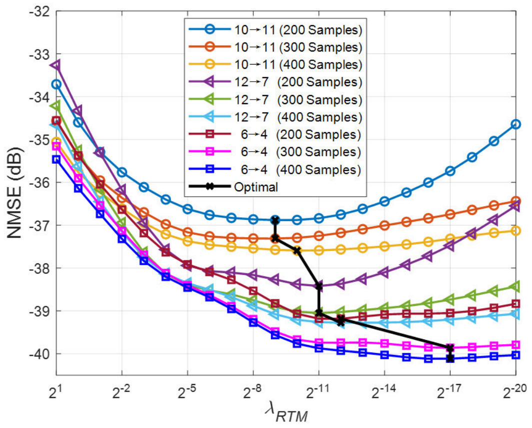


Fig. 12. NMSE results of RTM with different hyperparameters λ_{RTM}.

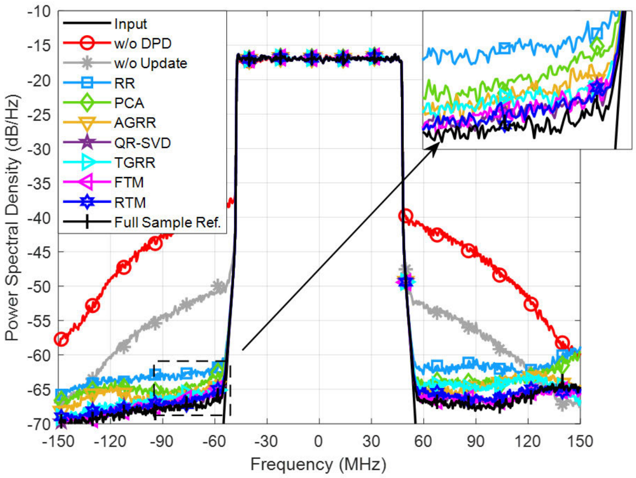


Fig. 13. PSD results comparison of various methods in state 11.

Although the optimal hyperparameter fluctuates and shows a decreasing trend as the “distance” between the source task and the target task increases. When $\lambda_{\text{RTM}} \approx 2^{-11}$, it is sufficient to obtain satisfactory results for the given situations, and the difference from the optimal performance is insignificant. Therefore, λ_{RTM} can be fixed in the majority of cases.

F. Experimental Results of DPD

To further validate the proposed methods, a DPD test was also performed. During all DPD tests, IDL was employed for estimating model coefficients.

Three sets of experiments were undertaken to strengthen the persuasiveness. Specifically, the source parameters in tests are the DPD coefficients of state 12 (100 MHz, −1 dBm), state 10 (100 MHz, −3 dBm), and state 6 (80 MHz, −3 dBm), which can be regarded as a priori knowledge. And the corresponding target tasks are to linearize the PA of configurations 7 (80 MHz, −2 dBm), 11 (100 MHz, −2 dBm), and 4 (60 MHz, −1 dBm). The DPD model and the setting of the source tasks for QR-SVD is consistent with that in forward modeling tests, with the number of coefficients $P = 186$, and 300 samples selected from the training set by MFB-OptiSim were used for parameter identification.

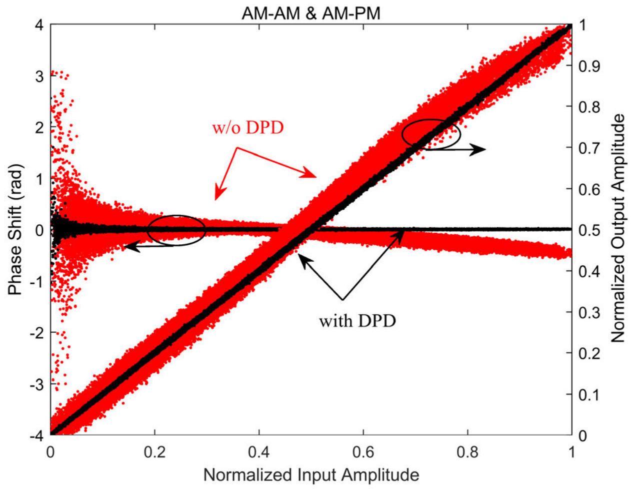


Fig. 14. AM/AM and AM/PM plots with and without DPD in state 11 with FTM for parameter extraction.

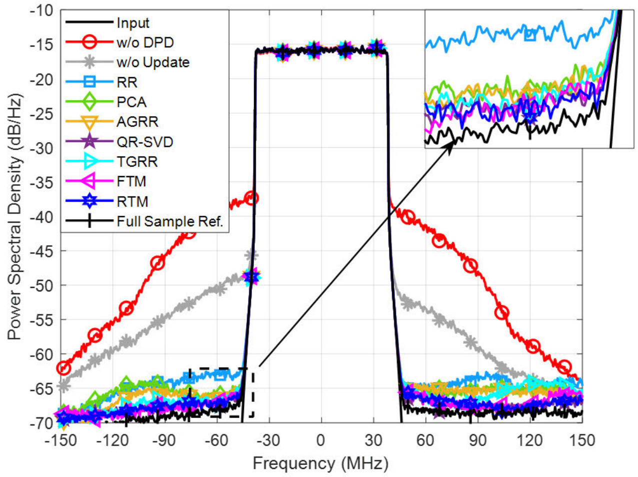


Fig. 15. PSD results comparison of various methods in state 7.

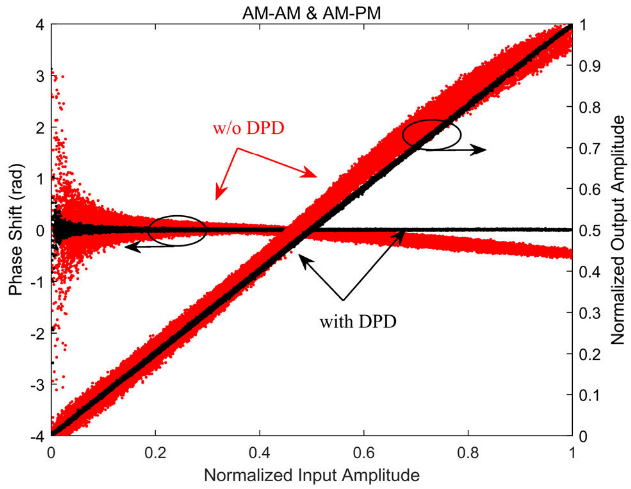


Fig. 16. AM/AM and AM/PM plots with and without DPD in state 7 with RTM for parameter extraction.

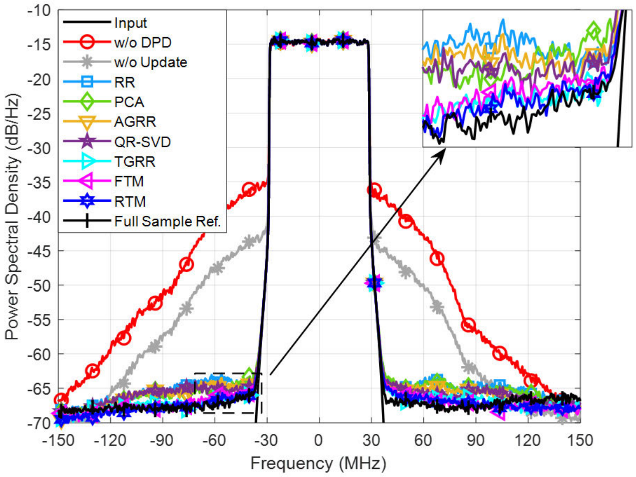


Fig. 17. PSD results comparison of various methods in state 4.

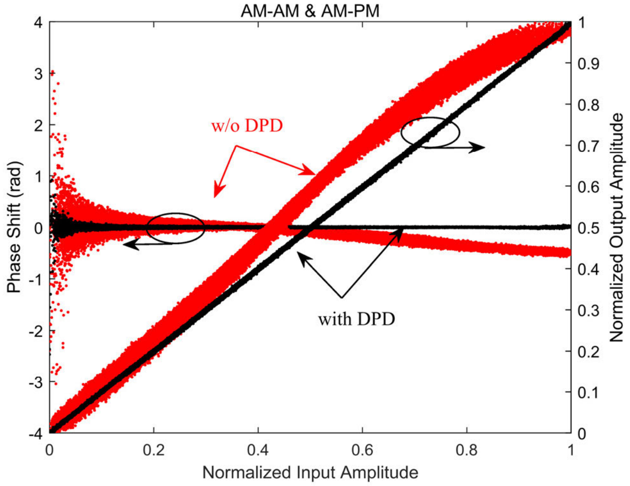


Fig. 18. AM/AM and AM/PM plots with and without DPD in state 4 with RTM for parameter extraction.

TABLE III

DPD Results of State 11

Method	NMSE (dB)	ACPR (dBc) (-/+ 10MHz)	ACPR (dBc) (-/+30 MHz)
w/o DPD	-17.29	-27.17/-29.65	-30.57/-33.12
w/o Update	-28.28	-38.88/-40.73	-42.08/-43.90
RR [24]	-38.89	-46.55/-44.21	-47.86/-44.99
PCA [27]	-40.43	-48.08/-45.63	-49.31/-46.26
AGRR [25]	-41.22	-48.98/-47.15	-50.20/-47.83
QR-SVD [5]	-42.27	-50.54/-48.72	-51.90/-49.55
TGRR	-41.77	-49.96/-47.44	-51.39/-48.45
FTM	**-42.36**	**-50.63/-48.97**	**-52.06/-49.88**
RTM	-42.18	-50.57/-48.51	-51.94/-49.40
Full Sample Ref.	-42.86	-51.30/-49.04	-52.58/-49.77

The measured power spectral density (PSD) of the three tests mentioned above is shown in Figs. 13, 15, and 17, respectively. In addition, the corresponding AM–AM and AM–PM characteristics by applying the proposed methods are drawn in Figs. 14, 16, and 18, and the detailed results are listed in Tables III–V, respectively. Although the proposed methods appear to be evenly matched with QR-SVD in configurations 11 and 7, the robustness of the proposed method may be better and the pretraining complexity is much lower, as stated in the analysis in Section V-B.

To better exploit the potential of the proposed methods, a more challenging scenario (state 13) is added for the

TABLE IV
DPD RESULTS OF STATE 7

Method	NMSE (dB)	ACPR (dBc) (-/+ 10MHz)	ACPR (dBc) (-/+30 MHz)
w/o DPD	-17.14	-27.29/-29.60	-31.58/-34.09
w/o Update	-28.23	-37.38/-40.92	-40.41/-44.27
RR [24]	-40.91	-48.06/-48.12	-49.22/-47.98
PCA [27]	-41.53	-49.21/-49.11	-49.59/-49.09
AGRR [25]	-41.90	-50.10/-49.13	-50.80/-49.32
QR-SVD [5]	**-42.98**	**-51.44/-51.17**	**-52.05/-51.25**
TGRR	-41.83	-50.98/-49.57	-51.64/-49.23
FTM	-42.77	-51.75/-51.06	-52.30/-50.91
RTM	-42.96	-51.51/-51.01	-52.09/-51.16
Full Sample Ref.	-43.64	-52.92/-52.31	-53.69/-52.44

TABLE V
DPD RESULTS OF STATE 4

Method	NMSE (dB)	ACPR (dBc) (-/+ 10MHz)	ACPR (dBc) (-/+30 MHz)
w/o DPD	-15.49	-26.68/-28.99	-32.37/-35.51
w/o Update	-22.57	-33.97/-36.42	-39.13/-42.51
RR [24]	-41.37	-49.60/-49.70	-49.70/-50.26
PCA [27]	-41.61	-50.38/-49.90	-50.94/-50.09
AGRR [25]	-41.63	-50.16/-50.28	-50.65/-50.69
QR-SVD [5]	-41.88	-50.30/-50.62	-50.60/-50.89
TGRR	-42.73	-51.94/-51.67	-52.39/-51.97
FTM	-42.72	-51.90/-51.74	-52.56/-52.38
RTM	**-42.83**	**-52.20/-51.85**	**-52.84/-52.31**
Full Sample Ref.	-43.77	-52.55/-52.61	-52.89/-52.52

TABLE VI
DPD RESULTS OF STATE 13

Method	NMSE (dB)	ACPR (dBc) (-/+ 10MHz)	ACPR (dBc) (-/+30 MHz)
w/o DPD	-12.90	-23.55/-24.45	-26.50/-27.62
w/o Update	-28.28	-37.32/-39.69	-39.06/-41.72
RR [24]	-35.04	-42.66/-42.30	-44.44/-43.99
PCA [27]	-35.99	-43.14/-43.43	-44.60/-44.83
AGRR [25]	-36.92	-44.36/-42.78	-45.75/-43.57
QR-SVD [5]	-36.37	-45.43/-42.54	-46.68/-44.01
TGRR	-37.34	-45.77/-43.90	-46.94/-45.03
FTM	-38.25	-45.91/-43.51	-47.31/-44.26
RTM	**-38.29**	**-45.66/-43.83**	**-47.05/-44.68**
Full Sample Ref.	-39.71	-47.56/-44.13	-48.77/-44.70

evaluation of linearization performance, with 0-dBm input power and 100-MHz bandwidth. Due to the more complex nonlinear characteristics, the number of training samples N_T is increased from 300 to 500. Following the principles mentioned in Section V-A, the source parameter is selected to the ideal DPD coefficients in state 12 for the proposed methods, and the source tasks for QR-SVD remain unchanged. Fig. 19 shows the spectral results of various methods, and the AM–AM and AM–PM characteristics by applying the proposed RTM are shown in Fig. 20. It is clear that the proposed methods achieve leading linearization performance in this case. The results are detailed in Table VI.

G. Performance of the Proposed Methods in Extreme Cases

To further analyze how the performance of the suggested strategy is affected by the reliability of the source task, the forward modeling results of the proposed method in extreme cases (13→1 and 1→13) are detailed in Table VII, which contains the settings of the hyperparameters for FTM and RTM. And the NMSE accuracy by applying LS, RR, AGRR, and PCA is also given as references, with 300 samples for training. It can be clearly seen that even in severe transfer circumstances, i.e., when the "distance" between the source task and the target task is large, which means the source parameters do not ideal enough, the proposed methods still exhibit some improvement over LS. Although it appears to offer no significant advantage above previous methods in those two extreme cases, at least the negative influence on parameter estimation is avoided, which further demonstrates the robustness of the proposed method in practical applications.

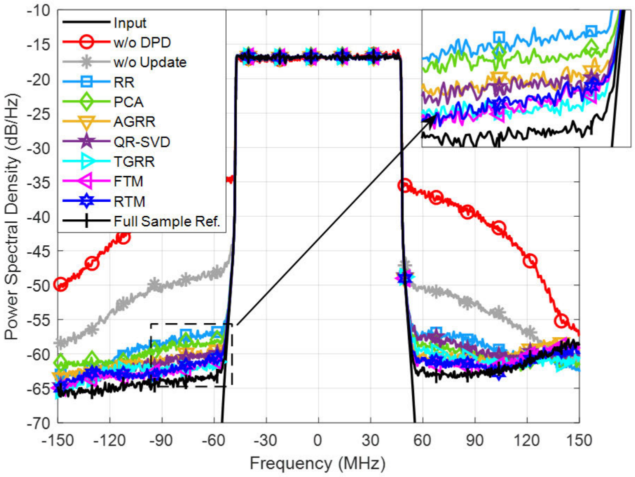


Fig. 19. PSD results comparison of various methods in state 13.

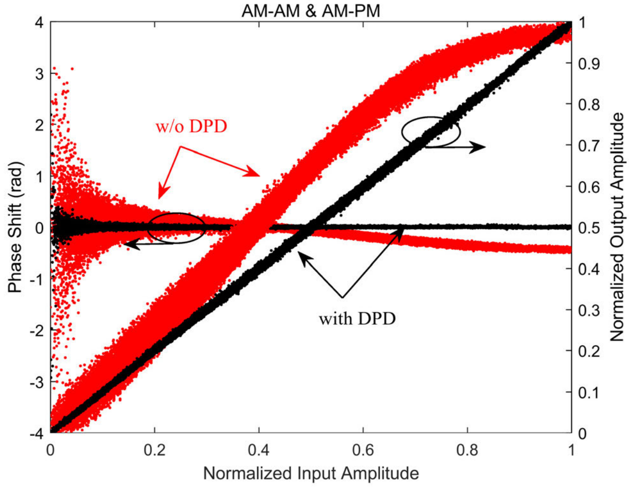


Fig. 20. AM/AM and AM/PM plots with and without DPD in state 13 with RTM for parameter extraction.

Combining the results above, the proposed methods are demonstrably useful and robust to reduce the estimation error of FSL. Please note that the proposed methods are generic, so they could be utilized in a variety of situations to improve the robustness of parameter estimation, such as band-limited feedback, I/Q imbalance, and group delay distortion. In addition, for the application of multiple PAs, such as multiple-input–multiple-output (MIMO) systems, the proposed methods can also be applied since the nonlinear characteristics of each PA are similar.

TABLE VII
FORWARD MODELING RESULTS IN EXTREME CASES

Method	13→1	1→13
w/o Update	-19.30	-17.63
LS	-37.08	-33.27
RR [24]	-40.00	-33.66
AGRR [25]	-40.12	-35.12
PCA [27]	**-40.61**	-35.34
TGRR	-40.29	**-35.62**
FTM (μ & $iter$)	-39.97 ($\mu = 10^{-6}$) ($iter = 10^{9.5}$)	-35.18 ($\mu = 10^{-7}$) ($iter = 10^{11}$)
RTM (λ_{RTM})	-39.95 ($\lambda_{RTM} = 2^{-20}$)	-35.22 ($\lambda_{RTM} = 2^{-19}$)
Full Sample Ref.	-41.67	-37.26

It is worth mentioning that the approaches presented in this article are based on a fixed model structure (homogeneous transfer learning), i.e., the same DPD model is employed for all states, which may lack some flexibility. Obviously, the case of weak nonlinearity does not require a high model order (the configuration of low input power or narrowband transmission signal). We believe that transfer learning between different model structures (heterogeneous transfer learning) will be one of the future trends [36]. Furthermore, source parameters should not be restricted to coefficients of a certain state. Applying a weighted fusion of known multistate parameters might be a better choice. In conclusion, the approaches introduced in this article are effective and promising, leading to new possibilities for low-complexity extraction of DPD model coefficients.

VI. CONCLUSION

In this article, we proposed three transfer learning-based parameter identification methods to reduce the error in DPD model coefficients estimation. Different from the earlier strategies which merely initialize, the suggested methods exploit the structural information of past parameters to impose reasonable constraints on parameter estimation, and thus substantially reduce the bias. Besides, the proposed approaches have been derived as one-step noniterative solutions and are therefore easier to deploy and of limited complexity. Whether TGRR and FTM, which require pretraining (for TGRR is the orthogonal transformation matrix $\mathbf{U}$, and for FTM is the transfer matrix $\mathbf{\Phi}^{\text{iter}}$), or RTM, which is pretraining free, their additional computational complexity in parameter extraction is insignificant compared to standard LS. The experimental results demonstrate that the proposed methods outperform previous state-of-the-art methods, which provide a new idea for the rapid adaptation of DPD.

REFERENCES

[1] L. Guan and A. Zhu, "Green communications: Digital predistortion for wideband RF power amplifiers," *IEEE Microw. Mag.*, vol. 15, no. 7, pp. 84–99, Nov. 2014.

[2] S. C. Cripps, *RF Power Amplifiers for Wireless Communications*, 2nd ed. Norwood, MA, USA: Artech House, 2006.

[3] G. Jindal, G. T. Watkins, K. Morris, and T. A. Cappello, "Digital predistortion of RF power amplifiers robust to a wide temperature range and varying peak-to-average ratio signals," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 7, pp. 3675–3687, Jul. 2022.

[4] N. Hammler, A. Cathelin, P. Cathelin, and B. Murmann, "A spectrum-sensing DPD feedback receiver with 30× reduction in ADC acquisition bandwidth and sample rate," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 9, pp. 3340–3351, Sep. 2019.

[5] Y. Li, X. Wang, and A. Zhu, "Complexity-reduced model adaptation for digital predistortion of RF power amplifiers with pretraining-based feature extraction," *IEEE Trans. Microw. Theory Techn.*, vol. 69, no. 3, pp. 1780–1790, Mar. 2021.

[6] Y. Li and A. Zhu, "On-demand real-time optimizable dynamic model sizing for digital predistortion of broadband RF power amplifiers," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 7, pp. 2891–2901, Jul. 2020.

[7] Y. Guo, C. Yu, and A. Zhu, "Power adaptive digital predistortion for wideband RF power amplifiers with dynamic power transmission," *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 11, pp. 3595–3607, Nov. 2015.

[8] J. Kim and K. Konstantinou, "Digital predistortion of wideband signals based on power amplifier model with memory," *Electron. Lett.*, vol. 37, no. 23, pp. 1417–1418, Nov. 2001.

[9] D. R. Morgan, Z. Ma, J. Kim, M. G. Zierdt, and J. Pastalan, "A generalized memory polynomial model for digital predistortion of RF power amplifiers," *IEEE Trans. Signal Process.*, vol. 54, no. 10, pp. 3852–3860, Oct. 2006.

[10] A. Zhu, J. C. Pedro, and T. J. Brazil, "Dynamic deviation reduction-based Volterra behavioral modeling of RF power amplifiers," *IEEE Trans. Microw. Theory Techn.*, vol. 54, no. 12, pp. 4323–4332, Dec. 2006.

[11] A. Zhu, "Decomposed vector rotation-based behavioral modeling for digital predistortion of RF power amplifiers," *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 2, pp. 737–744, Feb. 2015.

[12] J. Kral, T. Gotthans, R. Marsalek, M. Harvanek, and M. Rupp, "On feedback sample selection methods allowing lightweight digital predistorter adaptation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 6, pp. 1976–1988, Jun. 2020.

[13] T. Wang and P. L. Gilabert, "Mesh-selecting for computational efficient PA behavioral modeling and DPD linearization," *IEEE Microw. Wireless Compon. Lett.*, vol. 31, no. 1, pp. 37–40, Jan. 2021.

[14] G. Yang et al., "Digital predistortion based on sample selection with memory effect," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 32, no. 2, Feb. 2022, Art. no. e22976.

[15] G. Yang et al., "Memory feature-based sample selection strategy for few-sample learning digital predistortion," *IEEE Trans. Microw. Theory Techn.*, vol. 71, no. 2, pp. 602–612, Feb. 2023.

[16] R. N. Braithwaite, "Closed-loop digital predistortion (DPD) using an observation path with limited bandwidth," *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 2, pp. 726–736, Feb. 2015.

[17] Y. Liu, W. Pan, S. Shao, and Y. Tang, "A new digital predistortion for wideband power amplifiers with constrained feedback bandwidth," *IEEE Microw. Wireless Compon. Lett.*, vol. 23, no. 12, pp. 683–685, Dec. 2013.

[18] Y. Liu, W. Pan, S. Shao, and Y. Tang, "A general digital predistortion architecture using constrained feedback bandwidth for wideband power amplifiers," *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 5, pp. 1544–1555, May 2015.

[19] C. Jiang et al., "A manifold regularization approach for low sampling rate digital predistortion with band-limited feedback," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 11, pp. 4928–4939, Nov. 2022.

[20] K. Tang, C. Yu, S. Li, M. Su, and Y. Liu, "A low sampling rate memory-grouped method for digital predistortion with constrained acquisition bandwidth," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 1, pp. 476–489, Jan. 2022.

[21] Y. Liu, C. Huang, X. Quan, P. Roblin, W. Pan, and Y. Tang, "Novel linearization architecture with limited ADC dynamic range for green power amplifiers," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3902–3914, Dec. 2016.

[22] H. Wang, G. Li, C. Zhou, W. Tao, F. Liu, and A. Zhu, "1-bit observation for direct-learning-based digital predistortion of RF power amplifiers," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 7, pp. 2465–2475, Jul. 2017.

[23] L. Guan and A. Zhu, "Optimized low-complexity implementation of least squares based model extraction for digital predistortion of RF power amplifiers," *IEEE Trans. Microw. Theory Techn.*, vol. 60, no. 3, pp. 594–603, Mar. 2012.

[24] Y. Liu et al., "Relaxing requirements on training samples in digital predistortion by using ridge regression," *IEEE Microw. Wireless Compon. Lett.*, vol. 31, no. 6, pp. 616–619, Jun. 2021.

[25] G. Yang, W. Qiao, C. Jiang, L. Su, and F. Liu, "Generalized ridge regression-based few-sample learning digital predistortion," *IEEE Microw. Wireless Compon. Lett.*, vol. 32, no. 6, pp. 603–606, Jun. 2022.

[26] J. A. Becerra, M. J. M. Ayora, J. Reina-Tosina, and C. Crespo-Cadenas, "Sparse identification of Volterra models for power amplifiers without pseudoinverse computation," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 11, pp. 4570–4578, Nov. 2020.

[27] D. Lopez-Bueno, Q. A. Pham, G. Montoro, and P. L. Gilabert, "Independent digital predistortion parameters estimation using adaptive principal component analysis," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 12, pp. 5771–5779, Dec. 2018.

[28] R. Braithwaite, "A self-generating coefficient list for machine learning in RF power amplifiers using adaptive predistortion," in *Proc. Eur. Microw. Conf.*, Sep. 2006, pp. 1229–1232.

[29] D. Obst, B. Ghattas, S. Claudel, J. Cugliari, Y. Goude, and G. Oppenheim, "Improved linear regression prediction by transfer learning," *Comput. Statist. Data Anal.*, vol. 174, Oct. 2022, Art. no. 107499.

[30] W. J. Hemmerle, "An explicit solution for generalized ridge regression," *Technometrics*, vol. 17, no. 3, pp. 309–314, Aug. 1975.

[31] W. J. Hemmerle and T. F. Brantle, "Explicit and constrained generalized ridge estimation," *Technometrics*, vol. 20, no. 2, pp. 109–120, May 1978.

[32] M. Ohishi, H. Yanagihara, and Y. Fujikoshi, "A fast algorithm for optimizing ridge parameters in a generalized ridge regression by minimizing a model selection criterion," *J. Stat. Planning Inference*, vol. 204, pp. 187–205, Jan. 2020.

[33] W. N. van Wieringen, "Lecture notes on ridge regression," 2015, *arXiv:1509.09169*.

[34] A. S. Tehrani, H. Cao, S. Afsardoost, T. Eriksson, M. Isaksson, and C. Fager, "A comparative analysis of the complexity/accuracy tradeoff in power amplifier behavioral models," *IEEE Trans. Microw. Theory Techn.*, vol. 58, no. 6, pp. 1510–1520, Jun. 2010.

[35] C. Eun and E. J. Powers, "A new Volterra predistorter based on the indirect learning architecture," *IEEE Trans. Signal Process.*, vol. 45, no. 1, pp. 223–227, Jan. 1997.

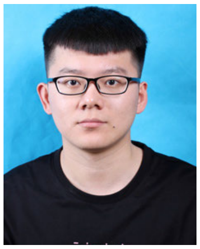
[36] O. Day and T. M. Khoshgoftaar, "A survey on heterogeneous transfer learning," *J. Big Data*, vol. 4, no. 1, pp. 1–42, Sep. 2017.

Guichen Yang received the B.E. degree from the School of Electronics and Information Engineering, Hefei University of Technology (HFUT), Hefei, China, in 2019. He is currently pursuing the Ph.D. degree in electromagnetic field and microwave technology at the University of Science and Technology of China (USTC), Hefei.

His research interests focus on behavioral modeling and digital predistortion for radio frequency (RF)power amplifiers.

Chengye Jiang (Graduate Student Member, IEEE) received the B.E. degree from the Department of Electronic and Information Engineering, Civil Aviation University of China, Tianjin, China, in 2019. He is currently pursuing the Ph.D. degree in electromagnetic field and microwave technology at the University of Science and Technology of China, Hefei, China.

His current research interests include digital predistortion, nonlinear system modeling, and machine learning.



Renlong Han (Graduate Student Member, IEEE) received the B.E. degree in integrated circuit design and integrated systems from the Hefei University of Technology (HFUT), Hefei, China, in 2020. He is currently pursuing the M.E. degree in electromagnetic field and microwave technology at the University of Science and Technology of China (USTC), Hefei.

His research interests include digital predistortion, nonlinear system modeling, and machine learning.

Jingchao Tan received the B.E. degree in electronics science and technology from the University of Science and Technology of China (USTC), Hefei, China, in 2020, where he is currently pursuing the M.E. degree in information and communication engineering.

His research interests include MIMO transmitter modeling and digital predistortion.

Falin Liu was born in Xingtai, China, in 1963. He received the B.E. degree from Tsinghua University, Beijing, China, in 1985, and the M.E. and Ph.D. degrees in electronic engineering from the University of Science and Technology of China (USTC), Hefei, China, in 1988 and 2004, respectively.

From 1997 to 1998, he was a Visiting Scholar with Tohoku University, Sendai, Japan. Since 1988, he has been with the Department of Electronic Engineering and Information Science, USTC, where he is currently a Full Professor. He has authored more than 90 papers in refereed journals and international conferences. His current research interests include millimeter wave transceivers, computational electromagnetics, microwave devices and communications, and radar imaging.

Dr. Liu is a Senior Member of the Chinese Institute of Electronics, Beijing. He was a recipient of the Second Prize of the National Science and Technology Progress Award and the First Prize of the CAS Science and Technology Progress Award. He is an Associate Editor-in-Chief of the *Journal of Microwaves* (in Chinese).