Automatic Hybrid-Precision Quantization for MIMO Detectors

Yingmeng Ge, Zhenhao Ji, Yongming Huang, Senior Member, IEEE, Zaichen Zhang, Senior Member, IEEE, Xiaohu You, Fellow, IEEE, and Chuan Zhang, Senior Member, IEEE

Abstract—In the design of wireless systems, quantization plays a critical role in hardware, which directly affects both area efficiency and energy efficiency. Being an enabling technique, the wide applications of multiple-input multiple-output (MIMO) heavily rely on efficient implementations balancing both performance and complexity. However, most of the existing detectors quantize all variables in a uniform way, resulting in high redundancy and low flexibility. Requiring both expertise and efforts, an in-depth tailored quantization usually asks for prohibitive costs and is not considered by conventional MIMO detectors. In this paper, a general framework named the automatic hybrid-precision quantization (AHQP) is proposed with two parts: integral part quantization (IPQ) determined by probability density function (PDF), and fractional part quantization (FPQ) by deep reinforcement learning (DRL). Being automatic, AHQP demonstrates high efficiency in figuring out good quantizations for a set of algorithmic parameters. For the approximate message passing (AMP) detector, AHQP achieves up to 57.8% lower average bitwidth than the unified quantization (UQ) one with almost no performance sacrifice. The feasibility of AHQP has been verified by implementation with 65 nm CMOS technology. Compared with its UQ counterpart, AHQP exhibits 2.97× higher throughput-to-area ratio (TAR) with 19.3% lower energy dissipation. Moreover, by node compression and strength reduction, the AHQP-guided detector outperforms the state-of-the-art (SOA) in both throughput (17.92 G/s) and energy efficiency (7.93 pJ/b). The proposed AHQP framework is also applicable for other digital signal processing (DSP) algorithms.

Index Terms—Automatic quantization, MIMO detection, message passing, DRL, ASIC.

I. INTRODUCTION

WITH the increasing demands of wireless communications, multiple-input multiple-output (MIMO) has gained wide attentions due to its high spectral efficiency (SE) and energy efficiency (EE) [1]. To alleviate the unbearable complexity of the optimal MIMO detectors, Bayesian message passing (BMP) algorithms have been widely considered to better balance performance and complexity, including belief propagation (BP), channel hardening-exploiting message passing (CHEMP), approximate message passing (AMP), expectation propagation (EP), etc. To bridge the gap between algorithms and implementations, hardware of those algorithms has been explored. In [2], Peng et al. presented a pipelined high-throughput ASIC implementation of BP detector. Implementations of message passing detectors (MPDs) based on CHEMP were given in [3], [4]. Those MPDs were further improved by deep neural network (DNN) towards higher efficiency and flexibility [5], [6], [7] successfully reduced the complexity of EP detectors [8] with Neumann-series approximations.

Although the aforementioned detectors raised the area and energy efficiency at the expense of performance, the main focus was on the arithmetic implementations but not on the quantization optimizations. As the very first and important step from algorithms (floating-point) towards implementations (fixed-point), the central issue of quantization optimization is to achieve the best trade-off between performance and complexity. Usually, the complexity is proportional to quantization length, but there is no such linearity in the relationship between performance and quantization. Although shorter quantization will penalize the performance, increasing quantization is not always cost-effective considering the performance saturation. In addition, for each iteration of BMP detection, different variables have different requirements on quantization precision (intra-iteration). Those requirements also vary in different iterations (inter-iteration). Therefore, instead of figuring out an exactly-matching quantization, a unified quantization (UQ) for all variables, which is determined by limited numerical results and expertise, is usually adopted for implementation feasibility.

Saving design efforts, such UQ for all variables suffers from bitwidth redundancy and hardware overhead. Hence, hybrid-precision quantization with low design cost is highly expected by MIMO detectors. According to [9], this optimization can be categorized as a problem difficult to model, and AI techniques are expected to be promising helpers. Since the inter-iteration quantization can be transferred to the intra-iteration quantization by unfolding [10], only the latter will be discussed here.

In this paper, deep reinforcement learning (DRL) is utilized for automatic quantization. Though there are works on
DRL-based MIMO detections [11], [12], they mainly focus on algorithms. It is true that DRL-based quantization has been investigated for convolutional neural networks (CNNs) [13], [14], [15], its application for MIMO detection is quite different for the following reasons.

1) The structure of CNN is regular, mainly composed of convolutional layers, pooling layers, and fully connected layers. However, the MIMO detectors are diverse and without a universal structure. Taking BMP detectors as an example, EP, BP, and AMP are based on different factor graphs. Thus different BMP detectors vary in structures, bringing much more challenges.

2) The accuracy requirement of MIMO detectors is much more demanding than that of CNNs. For example, the top-5 error rate of the 16-bit ResNet-50 is 7.82% [15]. However, the required bit-error-rate (BER) of MIMO detector is usually lower than $10^{-5}$. Therefore, MIMO detectors are more sensitive to quantization noise.

3) The aforementioned DRL-based automatic quantization frameworks for CNNs are all layer-wise (or kernel-wise). However, for MIMO detectors all variables require different quantizations to maximally squeeze the hardware.

This paper devotes itself in proposing an automatic hybrid-precision quantization (AHPQ) for MIMO detectors. The contributions of this paper are listed as follows.

1) For complexity consideration, the AHPQ works in two parts: integral part quantization (IPQ) based on probability density function (PDF), and fractional part quantization (FPQ) based on DRL. The general solution of IPQ is obtained by leveraging Monte Carlo simulations. For DRL-based FPQ, definitions of state, action, and reward are given. Two approaches addressing the reward misjudgment are proposed for better performance.

2) A specific case, namely the quantization of AMP with nearest-neighbor approximation (NNA-AMP), is analyzed. The impact of the number of Monte Carlo samples, the number of extracted variables, and influence range of action is discussed with simulations. Compared with UQ, AHPQ presents a much lower quantization bitwidth. Also, Pareto analysis is provided.

3) To validate the advantages of AHPQ from a hardware perspective, a hardware-friendly NNA-AMP (HF-AMP) detector by node compression (NC) and strength reduction is proposed. The VLSI architecture is given as well.

4) The ASIC implementation of HF-AMP is presented. The HF-AMP using AHPQ presents advantages in both area and energy efficiency compared with that using UQ. Moreover, HF-AMP using AHPQ outperforms the state-of-the-art (SOA) in both throughput (17.92 Gb/s) and energy efficiency (7.93 pJ/b).

The reminder of this paper is organized as follows. In Section II, the preliminaries of MIMO, DRL, and quantization are reviewed. In Section III, AHPQ is proposed with details. An application example of MIMO detectors is provided in Section IV. In Section V, the HF-AMP and its efficient VLSI architecture are presented. The ASIC implementations are given and compared in Section VI. Section VII concludes the entire paper.

**Notations:** $I_n$ denotes the $n \times n$ identity matrix. $X^T$ is the transpose operation of the matrix $X$. $\mathcal{N}(\mu, \Sigma)$ denotes the multi-variate Gaussian distribution with mean vector $\mu$ and covariance matrix $\Sigma$, while $\mathcal{CN}(\mu, \Sigma)$ denotes the complex multi-variate Gaussian distribution with complex mean vector $\mu$ and complex covariance matrix $\Sigma$. Operation $\text{clip}(v, v_{\min}, v_{\max})$ is to clip value $v$ into range $[v_{\min}, v_{\max}]$. $\text{round}(v)$ function returns a integer number that is a rounded version of the specified number $v$. $\text{card}(\mathbf{S})$ returns the number of elements in the multiset $\mathbf{S}$. $\text{sign}(v)$ returns sign of $v$.

II. PRELIMINARIES

A. MIMO System Model

We consider a narrow-band MIMO communication system with $N_t$ transmitting (Tx) antennas and $N_r$ receiving (Rx) antennas. Suppose the modulation mode is $Q$-QAM with constellation $\mathcal{Q}$. The system model is as follows,

$$
y = \mathbf{H}x + \mathbf{n},$$

where $y \in \mathbb{C}^{N_r \times 1}$ and $x \in \mathbb{C}^{N_t \times 1}$ are the received and transmitted vectors, respectively. $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ is the channel matrix. The independent and identically distributed (i.i.d.) Rayleigh channel with mean zero and variance $1/N_r$ is assumed in this paper. $\mathbf{n} \sim \mathcal{CN}(0, \sigma_n^2 \mathbf{I}_{N_r})$ is the additive white Gaussian noise (AWGN) vector with mean zero and variance $\sigma_n^2$. We assume the channel state information (CSI) is perfectly known at the receiving end. The complex model is regularly transformed into an equivalent real model following [16],

$$
y = \mathbf{x} + \mathbf{n},$$

where $y \in \mathbb{R}^{2N_r \times 1}$, $\mathbf{x} \in \mathbb{R}^{2N_t \times 1}$, and $\mathbf{x} \in \Omega^{2N_t \times 1}$. $\Omega$ is the set of real/imaginary part of $Q$-QAM constellation with size of $\sqrt{Q}$. $\mathbf{n} \sim \mathcal{N}(0, \sigma_n^2 \mathbf{I}_{2N_r})$ in which $\sigma_n^2 = \sigma_n^2 / 2$.

B. Deep Reinforcement Learning

DRL, as a branch of machine learning (ML), is an unsupervised learning algorithm concerned with an agent interacting with the environment to make good decisions [17], possessing the power to handle the label-missing quantization problems. At timestep $t$, the agent receives a state $s_t \in \mathcal{S}$ by interacting with the environment and then selects an action $a_t \in \mathcal{A}$ according to policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$. The environment returns a reward $r_t = \mathcal{R}(s_t, a_t, s_{t+1})$ and the next state $s_{t+1}$ after accepting the agent’s action. The above process repeats until the environment returns termination, which is called an episode. Here, $\mathcal{S}$ and $\mathcal{A}$ indicate state space and action space, respectively. $\mathcal{R}$ is the reward function.

Proximal policy optimization (PPO) has become one of the most popular policy gradient algorithms due to its easy implementation and inexpensive computation. Trust region policy optimization (TRPO) is first proposed to solve the sample-inefficient problem in on-policy DRL algorithms by using importance sampling [18]. Later, PPO is proposed to improve the performance and reduce the complexity of TRPO by modifying...
the objective function [19]. For more detailed theories of PPO, we recommend that readers refer to [19].

C. Quantization

Linear quantization is adopted in this paper due to its efficient implementation on hardware. Specifically, the quantization scheme is as follows, the sign, integral part, and fractional part take 1, \( p \), and \( q \), respectively, abbreviated as \( 1 - p - q \). For a variable with the value of \( v \), the quantized value \( v_Q \) can be expressed as:

\[
v_Q = \text{round}(\text{clip}(v, B_{\min}, B_{\max})/C) \times C,
\]
where \( B_{\min} = -2^p \), \( B_{\max} = 2^p - 2^{-q} \) and \( C = 2^{-q} \).

III. THE PROPOSED AHPQ

For complexity consideration, we consider IPQ and FPQ to be conducted separately. Since the integral part bitwidth (IPB) can be determined by the PDF of the data, we can generate a large amount of data to obtain the statistics of variables.

Denote the data with \( N_{IPQ} \) samples generated by Monte Carlo simulation as a multiset \( S \). For \( 1 - p - q \) quantization scheme, the range of variables is \( [B_{\min}, B_{\max}] \). \( B_{\max} \) is related to the unknown \( q \) in the IPQ step because of the separate consideration of IPQ and FPQ. Thus we consider the worst case, which is \( q = 0 \), then \( B_{\max}' = 2^p - 1 \). The multiset composed of the elements in \( S \) that is not within the range \( [B_{\min}, B_{\max}] \) is denoted as \( S' \), and \( S' = \{ v | v < B_{\min} \text{ or } v > B_{\max}' \} \). The IPB can be determined as follows:

\[
p^* = \min p \quad \text{s.t.} \quad \frac{\text{card}(S')}{\text{card}(S)} \leq \varepsilon_1,
\]
where \( \varepsilon_1 \) is the threshold. When \( \varepsilon_1 = 0 \), \( \text{card}(S') = 0 \), meaning that the range of the variable can be covered in such quantization scheme.

As for FPQ, the optimal fractional part bitwidth (FPB) is usually determined by repeatedly testing BER performance under different FPBs, but it is time-consuming. In the following, we demonstrate the proposed DRL-based FPQ in terms of state, action, and reward function.

A. State

The state is defined as a vector \( (k, q_k) \), where \( k \in \{1, 2, \ldots, N_{\text{all}}\} \) is the \( k \)-th variable to be quantized. \( N_{\text{all}} \) denotes the number of variables to be quantized. \( q_k \in \{0, 1, \ldots, q_{\max}\} \) is the FPB of the \( k \)-th variable and \( q_{\max} \) denotes the preset maximum value of \( q_k \). Considering that the two-dimensional state may have poor learning ability for detection algorithms with many variables, we instead use one-hot encoding for \( k \) and \( q_k \). Thus the state becomes \( \text{ohe}[k], \text{ohe}[q_k] \), where operation \( \text{ohe}[\cdot] \) returns the one-hot encoding value. Then the dimension of state becomes \( N_{\text{all}} + q_{\max} + 1 \).

B. Action

The action \( a_t \) indicates the amount of change in FPB. Inspired by the circular queue mechanism, we use the following formula of FPB change,

\[
q_k' = q_k + a_t \mod (q_{\max} + 1),
\]
where \( q_k' \) is the FPB after taking action. The reason we do not clip \( q_k' \) into range \([0, q_{\max}]\) is that clipping operation will increase the probability of the agent getting stuck at 0 or \( q_{\max} \) (values less than 0 or greater than \( q_{\max} \) are forced to be 0 or \( q_{\max} \), respectively). After the agent takes action \( a_t \), the FPB of \( k \)-th variable is changed to \( q_k' \).

We define the influence range of an action as the maximum absolute value of bitwidth change at an action, referred to as \( L_a \). Thus, the action space \( \mathcal{A} \) is the set \( \{a | -L_a \leq a \leq L_a, a \in \mathbb{Z} \} \). The schematic diagram of action space with different \( L_a \) is presented in Fig. 1. When \( L_a = 1 \), the bitwidth can only be changed by at most 1 b, which may cause low learning efficiency and easy to fall into local optima. When \( L_a = \lfloor q_{\max}/2 \rfloor \), the agent can flexibly change current bitwidth to any other bitwidth. However, this may lead to instability in learning process and deterioration in convergence performance.

C. Reward Function

Due to the strict performance requirements in MIMO detection, the reward function in our DRL-based FPQ aims at reducing the FPB under the premise of ensuring BER performance.

We first focus on the BER performance evaluation. Since the precise BER requires a large number of samples, referred to as \( N_{\text{FPQ}} \), for Monte Carlo simulation to approach, it can be pre-computed before the agent starts learning, which is referred to as \( P_b \). Every time the environment functions, a small number of samples, referred to as \( N_{\text{FPQ}} \), is used to evaluate the BER performance of the floating-point detector and quantized detector simultaneously to save the computation time. The corresponding BER of the floating-point detector and the quantized detector is written as \( P_{b}^{\text{FL}} \) and \( P_{b}^{Q} \), respectively. The relative error of the BER of quantized detector is defined as \( (P_{b}^{Q} - P_{b}^{\text{FL}})/P_{b}^{\text{FL}} \). Considering that \( P_{b}^{\text{FL}} \) can be equal to 0 due to the small number of simulation samples, we instead use \( (P_{b}^{Q} - P_{b}^{\text{FL}})/P_{b} \) as the relative error. If the relative error is greater than threshold \( \varepsilon_2 \), the performance of the quantized detector cannot be guaranteed and the environment returns the reward \( r_t = -1 \). Otherwise, the quantization bitwidth is taken into consideration.

Denote the average FPB of the current quantization scheme as \( \bar{q} \). The reward function when relative error is smaller than \( \varepsilon_2 \) is defined as \( r_t = \theta_1 \exp(-\theta_2 \bar{q}/q_{\max}) \), where \( \theta_1 \) and \( \theta_2 \) are
the empirical parameters. Compared with a linear function, the exponential function can make the environment return a larger reward value when bitwidth is small, causing the agent to be more inclined to reduce the quantization bitwidth. To sum up, the reward function is as follows,

\[ r_t = \begin{cases} 
\theta_1 \exp(-\theta_2 q / q_{\text{max}}), & (P_b^Q - P_b^{FL}) / P_b \leq \varepsilon_2, \\
-1, & \text{otherwise}
\end{cases} \] (6)

Remark 1: Since the requirements for quantization precision at different signal-to-noise ratios (SNRs) are usually not consistent, the final quantization result cannot meet the BER requirements if BER performance is evaluated in the entire target range of SNRs. Instead, we can divide the whole target SNR interval into small sub-intervals or choose several tested SNR points. In each sub-interval (or at each point), the corresponding quantization scheme can be obtained separately using DRL-based FPQ, and the highest quantization precision is chosen as the final quantization scheme.

D. Reward Misjudgment Problem

Influenced by the mutual restriction of quantization precision of different variables to be quantized in a MIMO detector, considering all variables in one episode will introduce severe reward misjudgment and cause poor quantization results. Consider a specific case shown in Fig. 2, where Env., var., cur., and idl. denote environment, variable number, current FPB, and ideal FPB, respectively. Suppose the ideal quantization of the two variables var.1 and var.2 is var.1: 1−1−6 and var.2: 1−1−4, and the current quantization of the two variables is var.1: 1−1−6 and var.2: 1−1−6. The agent takes an action \( a_1 = -2 \) according to the assumed current state (1,6), and the quantization of var.1 becomes 1−1−4. The environment returns \( r_1 = -1 \) and selects (2, 6) as the next state. In this case, no matter what action the agent takes, the environment must return −1 since the quantization of var.1 does not meet the requirements. The reward misjudgment will become more serious for more variables. Therefore, two targeted approaches are proposed to solve the problem.

1) Extract Variables: We extract \( N_{\text{ext}} (N_{\text{ext}} \leq N_{\text{all}}) \) variables randomly that can be adjusted by the agent and other variables are fixed to maximum FPB in each episode. Then the average FPB is defined as follows,

\[ \bar{q} = \frac{1}{N_{\text{ext}}} \sum_{k \in S_{\text{ext}}} q_k, \] (7)

where \( S_{\text{ext}} \) denotes the set of extracted variables. Operating \( N_{\text{ext}} \) extracted variables in one episode can alleviate the reward misjudgment problem, and the restriction of quantization precision of all variables can be considered as long as the number of episodes is sufficient. When \( N_{\text{ext}} = 1 \), the BER performance is usually unsatisfactory due to lack of consideration for the restriction of quantization precision of variables.

2) Random Selection: After the environment accepting the action, the next variable is chosen randomly from \( S_{\text{ext}} \). Introducing more randomness to the learning process can reduce the probability of reward misjudgment problem.

E. Overall FPQ Process

In one episode, first, \( N_{\text{ext}} \) variables are randomly extracted. Then the agent interacts with environment until reaching the maximum timestep (\( T_{\text{max \_timestep}} \)). The state, action, and reward are collected and stored in a memory. After several episodes (\( T_{\text{update}} \)), the agent modifies the policy network and value network based on the experience from the memory using PPO algorithm. Once reaching the maximum episode (\( T_{\text{max \_episode}} \)), the agent stops learning.

Since the policy network cannot reflect the statistical characteristics of the FPB, Monte Carlo simulation is needed to obtain them. Fixing the \( k \)-th variable, the agent after learning keeps taking actions to change the FPB until reaching the maximum testing time (\( T_{\text{test}} \)). The mean of FPBs is the estimation of the expectation of \( q_k \).

To sum up, the detailed procedure of the proposed AHPQ is listed in Algorithm 1.

Remark 2: The detailed comparison between our proposed AHPQ and other DRL-based CNN quantization, including HAQ [13], ReLeQ [14], and AutoQ [15] are listed in Table I.

IV. AN APPLICATION EXAMPLE

A. NNA-AMP Detector

AMP, one popular BMP algorithm, has been widely considered in MIMO detection for a good trade-off between performance and complexity. The AMP detector is summarized...
in Algorithm 2, where \( b = H^T y \) denotes the received vector after matched filter, \( G = H^T H \) is the Gram matrix, \( b_i \) is the \( i \)-th element of \( b \), and \( g_{i,j} \) is the \((i,j)\)-th element of \( G \). \( E_s \) denotes the mean symbol energy. \( [\omega_1, \ldots, \sqrt{Q}] \) is the elements of \( \Omega \) in ascending order. \( u_i^{(l)} = \sum_{m=1}^{\sqrt{Q}} \exp[\Delta_i^{(l)}(\omega_m)] \) is the normalization coefficient. We recommend readers refer to [20], [21] for the detailed derivation and analysis of AMP algorithm.

For high order modulation, AMP detector suffers from unbearable computational complexity due to the moment matching (Lines 6-8 in Algorithm 2). Nearest-neighbor approximation (NNA) proposed in [4] simplifies the moment matching process by choosing the \( N_{\Omega} \) nearest neighbor symbols in constellation, respectively. In the following, AMP with \( N_{\Omega} = 2 \) NNA is adopted, referred to as NNA-AMP.

### Algorithm 2: AMP Algorithm.

**Input:** \( b, G, \sigma_n^2, L \)

**Output:** \( \tilde{x}_i^{(l)} (\forall i = 1, \ldots, 2N_t) \)

For \( l = 0, 1, \ldots, L - 1 \) do

for \( i = 1, 2, \ldots, 2N_t \) do

\[ z_i^{(l)} = \tilde{x}_i^{(l)} + d_i^{(l)} \]

\[ \tau_i^{(l)} = \sigma_n^2 + \beta_i^{(l)} \]

for \( m = 1, 2, \ldots, \sqrt{Q} \) do

\[ \alpha_i^{(l)}(\omega_m) = -2(\omega_m - z_i^{(l)})^2 + \tau_i^{(l)}) \]

\[ \Delta_i^{(l)}(\omega_m) = \alpha_i^{(l)}(\omega_m) - \max_{\omega_m^1, \omega_m^2} \alpha_i^{(l)}(\omega_m^1) \]

\[ \rho_i^{(l)}(\omega_m) = \frac{1}{\sqrt{2\pi}} \exp \left[ \frac{-\Delta_i^{(l)}(\omega_m^1)}{2} \right] \]

\[ \tilde{x}_i^{(l+1)} = \sum_{m=1}^{\sqrt{Q}} \omega_m \rho_i^{(l+1)}(\omega_m) \]

\[ \rho_i^{(l+1)}(\omega_m) = \frac{1}{\sqrt{2\pi}} \exp \left[ \frac{-\Delta_i^{(l)}(\omega_m)}{2} \right] \]

for \( i = 1, 2, \ldots, 2N_t \) do

\[ d_i^{(l+1)} = b_i - \sum_{j=1}^{2N_t} g_{i,j} \tilde{x}_j^{(l+1)} + \frac{\Delta_i^{(l+1)}}{2\sqrt{\tau_i^{(l)}}} d_i^{(l)} \]

### TABLE I

| Feature               | AHPQ | HAQ | RCI xQ | AutoQ |
|-----------------------|------|-----|--------|-------|
| separated IPQ and FPQ| √    | x   | x      | x     |
| flexible action space | √    | x   | x      | x     |
| hardware overhead estimator | ×0 | √ | ×      | ✓     |
| solving reward misjudgement | ✓ | × | ✓      | x     |

\* Due to the varied structure of detectors, the hardware overhead is hard to evaluate and will be explored in our further works.

### B. Implementation Details

Our simulation is performed in Pytorch 1.10.0 with AMD Ryzen™ 9 5950X CPU and NVIDIA Tesla P40.

We consider a massive MIMO system with \( N_t = 8 \) and \( N_r = 128 \). The modulation mode is 16-QAM. In NNA-AMP algorithm, all variables and their corresponding number are listed in Table II. The maximum iteration number \( L \) is set to 4 according to Fig. 3.

Fixed throughout the experiment, \( \varepsilon_1 = 0, \varepsilon_2 = 0.4, \theta_1 = 10, \) and \( \theta_2 = 4 \) can provide a good trade-off between BER performance and bitwidth after a certain simulation. \( q_{\text{max}} = 10 \). \( N_{\text{DPQ}} \) and \( N_{\text{FPQ}} \) are both set to 200000. \( T_{\text{max,episode}} = 30000 \), \( T_{\text{max, timestep}} = 40 \), and \( T_{\text{update}} = 150 \). BER performance evaluation is conducted at 6 dB because NNA-AMP needs the highest precision at 6 dB through plenty of trials.

Both policy and value networks consist of fully connected DNN with 6 hidden layers, whose dimensions are 64, 128, 256, 256, 128, 64, respectively. The dimensions of the input layers of both policy network and value network are equal to that of state space, which is 32 in our case. The dimension of the output layer of policy network is equal to that of action space, while the dimension of the output layer of the value network is 1.
C. Simulation Results

1) Effect of $N_{FPQ}^*$: We test the learning time, average FPB, and SNR loss with different $N_{FPQ}^*$, which is listed in Table III. As illustrated, the smaller $N_{FPQ}^*$ comes with lower time cost. When $N_{FPQ}^*$ is too small (500 or 2000), the resulting inaccurate BER calculations will make the agent in DRL difficult to converge and prevent the average FPB from reducing to the optima.

Generally, $N_{FPQ}^* = 4000$ can achieve a good trade-off between learning time, average FPB, and SNR loss. Considering only a single process is utilized in our algorithm implementation, the experience-collecting operation takes a lot of time, though it can be done in parallel. Therefore, better hardware platform, more efficient algorithm implementation, and multi-process operation can further reduce the training time, indicating that our proposed DRL-based FPQ has the potential in saving labor design efforts.

2) Effect of $L_a$: Fixing $N_{ext} = 5$, the rewards versus updates of policy with different $L_a$ is shown in Fig. 4(a). As illustrated, the convergence performance has a slight advantage when $L_a = 2$ compared with the case of $L_a = 1$. The reward value eventually converges to approximately the same value in both cases $L_a = 1$ and $L_a = 2$. When $L_a = 5$, the convergence rate becomes very slow, and the reward value at convergence is smaller than the other two cases. Therefore, the convergence rate and final FPB are unsatisfactory when $L_a$ is too large. In the following discussion, we fix $L_a$ to 2.

3) Effect of $N_{ext}$: The rewards returned by environment versus updates of policy with different $N_{ext}$ is shown in Fig. 4(b). As the number of updates increases, the reward value increases first and finally fluctuates around a certain value, indicating the agent has been learning to quantize variables in NNA-AMP detector better. Since the restriction of quantization precision of different variables is not taken into consideration, the reward value is large when $N_{ext} = 1$, indicating that the FPB of variables is reduced to a very low level. As $N_{ext}$ increases, the reward value at convergence becomes smaller. This is mainly because the more severe reward misjudgment problem makes the agent more challenging to reduce the FPB.

The BER performance comparison of AMP, NNA-AMP, and AHPQ-based quantized NNA-AMP with different $N_{ext}$ is shown in Fig. 5(a), where the legend entry “$N_{ext} = 1$” denotes AHPQ-based quantized NNA-AMP with $N_{ext} = 1$ and others are similar. As presented, the NNA-AMP algorithm can perfectly recover the performance of AMP. The quantized NNA-AMP shows degraded performance when $N_{ext} = 1$, while it presents a similar performance to the floating-point NNA-AMP when $N_{ext}$ is equal to 5, 15, and 21.

The trade-off between average bitwidth and performance loss after quantization is shown in Fig. 6(a). The horizontal axis stands for the SNR loss of NNA-AMP using AHPQ with the floating-point NNA-AMP as the benchmark. The vertical axis represents the average FPB. The quantized NNA-AMP when $N_{ext} = 5$ can allocate small quantization bitwidth while maintaining the performance of the original algorithm. Thus in the following, $N_{ext}$ is fixed to 5 and the detailed quantization bitwidth of AHPQ-based quantized NNA-AMP with $N_{ext} = 5$ is listed in Table IV.

4) Comparison With UQ Scheme: The IPB of UQ cannot be less than the maximum IPB of all variables, which is 7 bits. As for FPB, the BER performance comparison of NNA-AMP and NNA-AMP using UQ with different FPBs is presented in Fig. 5(b). The NNA-AMP using UQ suffers from severe performance deterioration when FPB is 4 or 5 bits, while it can recover the performance of floating-point NNA-AMP when FPB is 6 bits or more. Therefore, the UQ scheme of NNA-AMP is taken as $1 - 7 - 6$. The trade-off between average bitwidth and SNR loss of quantized NNA-AMP using AHPQ and UQ benchmarking with the floating-point NNA-AMP is presented in Fig. 6(b). As presented, AHPQ has only about 43.1% average bitwidth
Fig. 5. BER performance comparison ($N_t = 8$, $N_r = 128$, 16-QAM modulation) of (a) AMP, NNA-AMP, and AHPQ-based quantized NNA-AMP with different $N_{\text{ext}}$, (b) NNA-AMP and UQ-based quantized NNA-AMP with different quantization, and (c) the proposed HF-AMP and the SOA.

![Graph]  

TABLE IV
QUANTIZATION BITWIDTH COMPARISON OF NNA-AMP USING AHPQ AND UQ

| $k$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | Avg. |
|-----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|-----|
| Integral Part | AHPQ | 3 | 2 | 1 | 1 | 2 | 2 | 4 | 4 | 4 | 1 | 1 | 1 | 1 | 4 | 3 | 4 | 3 | 3 | 1 | 6 | 7 | 7.6 |
| UQ | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7.6 |
| Fractional Part | AHPQ | 6 | 7 | 3 | 3 | 3 | 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 1 | 4 | 4 | 6 | 4 | 0 | 1 | 1.28 |
| UQ | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |

Fig. 6. The trade-off between average FPB/bitwidth and SNR loss at BER $= 10^{-3}$, benchmarking with the floating-point NNA-AMP.

Fig. 7. Average FPB versus SNR loss space and its Pareto front for NNA-AMP quantization. The purple dashed line denotes the Pareto front, red star denotes our AHPQ scheme, green points (♦) denote the UQ schemes, and blue points (×) denote some other feasible hybrid-precision quantization schemes. The shaded area represents all feasible quantization schemes.

V. HARDWARE-FRIENDLY NNA-AMP DETECTOR AND VLSI ARCHITECTURE

A. Node Compression

As summarized in Table II, there are a large number of processing nodes in the signal flow graph (SFG) of NNA-AMP detector. Benefiting from AHPQ, the detailed quantization information of each variable makes it possible to compress the processing nodes by removing variables with low quantization bitwidth from the SFG.

Consider the operations related to variance updating in Lines 10–13 of Algorithm 2. Note that the quantization of $\xi^{(l)}$ is...
1 - 1 - 0 and \( \beta = N_t/N_r = 1/16 \), thus the value of \( \beta \xi^{(i)} \) must be less than 0.1. However, the AHPQ claims that the FPB of \( \beta \xi^{(i)} \) is 0, leading to this variable being set to 0 in the quantized NNA-AMP detector. Thus we can simplify Line 4 and Line 13 in Algorithm 2 as follows,

\[
\begin{align*}
\hat{\chi}_i^{(1)} &= \sigma_i^2, \\
\hat{d}_i^{(1+1)} &= b_i - \sum_j g_{i,j} \hat{z}_j^{(1+1)},
\end{align*}
\]

and the calculation of \( \hat{\xi}_i^{(1)} \) can be removed.

To verify the correctness of the aforementioned simplification, we simulate the BER performance of original NNA-AMP and NNA-AMP with NC under different antenna configurations. Fig. 8 demonstrates that the system performance only suffers negligible degradation when the antenna ratio \( N_t/N_r \) is smaller than 1/8. The variance term \( \xi^{(1)} \) dominates the performance loss when \( N_t = 32 \), thus forcing the bitwidth of \( \xi^{(1)} \) to 0 results in performance deterioration.

\section*{B. Strength Reduction}

In this subsection, according to the quantization results, we introduce a digital transformation to reorganize the digital signal processing (DSP) operations aiming at reducing the computational burden. In such a case, metrics can be efficiently achieved, such as small area, low power, and high throughput.

\subsection*{1) Simplification of Mean Calculation}

We analyze the calculation of \( \rho_i^{(1+1)}(\omega_m) \) in NNA-AMP with 16-QAM modulation. How to obtain the positions of the first and second nearest neighbor symbols in constellation via \( z_i^{(1)} \) affects the hardware design critically. Searching directly by Euclidean distance requires minus, square, and sorting operations, bringing intolerable latency and complexity.

Based on the coordinates of constellation, a selective branch structure is chosen to guide the hardware design, which can further simplify the dynamic multiplication associated with \( a_\omega \) and \( s_\omega \) in (9). As indicated in Fig. 9, index \( \{m_1, m_2\} \) divides the whole value interval of \( z_i^{(1)} \) into six parts as \( (-\infty, -2) \cup (-2, -1) \cup (-1, 0) \cup (0, 1) \cup (1, 2) \cup (2, +\infty) \) using Gray-coded flags \( \{F_1, F_2, F_3\} \) to distinguish. \( a_\omega \) and \( s_\omega \) can be pre-calculated in different intervals. \( a_\omega \) has three possible values for different intervals, namely \(-4, 0, 4\), and the flags \( \{F_4, F_5\} \) for choosing different values of \( a_\omega \) are Gray-coded as 01, 11, 10. \( s_\omega \) has two possible values with amplitudes of \(-2, 2\), respectively, and the flag \( F_6 \) for choosing different value of \( s_\omega \) is Gray-coded as 01. Hence, (9) can be simplified as

\[
\tilde{\Delta}_i^{(1)} = -2 \left| \text{sign}(s_\omega) \cdot \left( \chi_i^{(1)} - \frac{2}{\tau_i^{(1)}} \cdot \text{sign}(a_\omega) \right) \right|,
\]

where \(-| \cdot | \) and \(-\frac{2}{\tau_i^{(1)}} \cdot \text{sign}(a_\omega) \) can be obtained by numerical transformation and segment shift selector, respectively. The flag \( F_6 \) can be discarded since the calculation of \( \tilde{\Delta}_i^{(1)} \) is independent of \( s_\omega \) in (11). Furthermore, considering the calculation of \( \hat{\xi}_i^{(1+1)} \) (Line 9 of Algorithm 2), the multiplication with integer constellation can be simplified to shift-and-adder (SAA) operations by sub-expression sharing in Table V inspired by [22].

With \( F_6 \) discarded, five flags \( \{F_1, F_2, F_3, F_4, F_5\} \) which can be obtained by the input signal \( z_i^{(1)} \) directly guide the calculation of \( \hat{\Delta}_i^{(1)} \) and \( \hat{z}_i^{(1+1)} \).

\subsection*{2) Piecewise Linear Approximation}

We perform piecewise linear approximation (PLA) for nonlinear operations in NNA-AMP, conducting appropriate approximation in a limited numerical interval with negligible performance loss.
HF-AMP Algorithm.

Algorithm 3: HF-AMP Algorithm.

Input : \( b, G, \sigma_s^2, L \).

\[ r_0^{(0)} = b_i \hat{x}_i^{(0)} = E_t E_0^{(0)} = 0 (\forall i = 1, \ldots, 2N_t). \]

Output: \( \hat{x}_i^{(L)} (\forall i = 1, \ldots, 2N_t). \)

1. for \( l = 0, 1, \ldots, L - 1 \) do
2. for \( i = 1, 2, \ldots, 2N_t \) do
3. \( q_i^{(l)} = \hat{x}_i^{(l)} + \omega_i^{(l)}; \)
4. find \( m_1 \) and \( m_2; \)
5. calculate \( \Delta_i^{(l)} \) according to (11);
6. calculate \( \rho_i^{(l)} (\omega_m) \) in (8) with PLA;
7. calculate \( \hat{x}_i^{(l+1)} \) according to Table V;
8. \( \Delta_i^{(l+1)} = b_i - \sum_j g_{ij} \hat{x}_j^{(l+1)}; \)

The computational complexity of NNA-AMP detector is mainly focused on the calculation of \( \frac{1}{1 + \exp(\Delta_i^{(l)})} \). Considering that when the value of \( \Delta_i^{(l)} (\Delta_i^{(l)} < 0) \) is small enough, \( \exp(\Delta_i^{(l)}) \) can be approximately equal to 0. We first clip \( \Delta_i^{(l)} \) into range \([\eta_{a,1}, \eta_{b,1} = 0]\) and then use a piecewise linear function with \( N_{seg,1} \) uniform segments to approximate the operations \( \frac{1}{1 + \exp(\Delta_i^{(l)})} \) in range \([\eta_{a,1}, \eta_{b,1} = 0]\). The BER performance comparison of NNA-AMP and quantized NNA-AMP using different \( f_1(\eta_{a,1}, \eta_{b,1}, N_{seg,1}) \) is presented in Fig. 10(a).

The BER performance comparison of NNA-AMP and quantized NNA-AMP using different \( f_2(\eta_{a,2}, \eta_{b,2}, N_{seg,2}) \) is presented in Fig. 10(b). The simulation of BER performance with different \( f_2(\eta_{a,2}, \eta_{b,2}, N_{seg,2}) \) presents a similar BER performance compared to other detectors.

As for \( \frac{1}{1 + \sigma_i^2} \), the clipping value is \( \sigma_i^2 = 25 \). Therefore, we first clip \( \sigma_i^2 \) into range \([\eta_{a,2} = 1/2, \eta_{b,2} = 15/8]\) and then use a piecewise linear function with \( N_{seg,2} \) uniform segments to approximate the operations \( \frac{1}{1 + \sigma_i^2} \). The simulation of BER performance with different \( f_2(\eta_{a,2}, \eta_{b,2}, N_{seg,2}) \) presents a similar BER performance compared to other detectors.

Remark 3: Unlike the conventional implementation of nonlinear operations where all possible values are directly stored in read-only memory (ROM) for look-up tables (LUTs), using PLA only needs to store the interval, slope, and intercept. In such case, the number of values to be stored is notably reduced. The storage overhead can be effectively alleviated. Compared with traditional generation methods, e.g., coordinate rotation digital computer (CORDIC) algorithm, the processing latency is considerably shortened to the processing time of a multiplier and an adder.

The NNA-AMP with NC and strength reduction, referred to as HF-AMP, is summarized in Algorithm 3. Meanwhile, the quantization scheme of HF-AMP according to Table IV is listed in Table VI.

The BER performance comparison between our proposed AHPQ-guided quantized HF-AMP and other SOAs is illustrated in Fig. 5(c). The iteration numbers of MMSE-Wei [23], MPD [4, 24, 25], LAMA [26], and proposed HF-AMP/HF-AMP (AHPQ) are set to 3, 4, 4, 4, respectively. The HF-AMP (even quantized HF-AMP using AHPQ) presents a similar BER performance compared to other detectors.

C. Overall VLSI Architecture

The system configuration of HF-AMP is the same as Section IV. With high-throughput and low-complexity design, the overall architecture is demonstrated in Fig. 11, composed of two main processing elements (PEs) named constellation processing element (CPE) and parallel interference cancellation (PIC) in each pipeline iteration, a control unit (CU) for clock and input/output (I/O) control, and register banks for storing the I/O and auxiliary data.

1) CPE Module: The branch selection signal \( \{F_1, F_2, F_3, F_4, F_5\} \) and the probability \( \rho_i^{(l+1)} (\omega_{m_1}) \) and \( \rho_i^{(l+1)} (\omega_{m_2}) \) are determined in CPE module illustrated in Fig. 12, where
$z$ is calculated by \(w_1\) and \(w_2\). The two input variables \(w_1\) and \(w_2\) are calculated separately in the hybrid-precision adder tree (HPA). The operand \(z\) is extended by the HPA-7b+ module. In this unit, variable \(z_1\) in Line 3 of Algorithm 3 is calculated by hybrid-precision adder (HPA) of HPA-7b+ module. Note that the superscript \((l)\) of variables is omitted for convenience in Figs. 12–15. Since the quantization schemes of input variables \(d_i^{(l)}\), input variables \(z_i^{(l)}\), and output variables \(\tilde{z}_i^{(l)}\) are \(1 - 3 - 4\), \(1 - 2 - 2\), and \(1 - 4 - 4\), respectively, we consider to implement the two’s complement addition without overflow and use adders with smaller bitwidths in HPA-7b+ module. In detail, the HPA is designed as a 7-bit half-adder. The operand \(z_1^{(l)}\) \((\hat{z}^{(4:0)})\) and part of operand \(d_i^{(l)}\) \((d^{(7:2)})\) are firstly extended to 7 bits. After the 7-bit half-adder, the output is assigned to \(z[8:2]\). The rest bits of \(z_1\) \((z[1:0])\) is directly equal to \(d[1:0]\).

For calculating \(\chi^{(l)}_i\) in (11), a hybrid-precision multiplier (HPM) is employed in HPM-S8 module. The two input operands of HPM in HPM-S8 module are 9 bits \((1 - 4 - 4)\) and 6 bits \((1 - 4 - 1)\) respectively, and 8 bits \((6 - 1)\) are taken as the output.

To obtain \(m_1\) and \(m_2\), NNA-CASE module, including two multiplexers MUX-Ω and MUX-Δ, first extracts the integral part of the auxiliary variable \(\tilde{z}_i^{(l)}\) and then selects the corresponding flags. MUX-Ω determines \(F_2\) and \(F_3\), and MUX-Δ determines \(F_1\), \(F_2\), \(F_3\). After that, \(F_4\), \(F_5\) are utilized to determine how to calculate \(\tilde{\Delta}_{i,\text{imp}}^{(l)}\) as follows,

\[
\tilde{\Delta}_{i,\text{imp}}^{(l)} = \begin{cases} 
\chi_i^{(l)} + (\frac{1}{17} \ll 1), & \{F_4, F_5\} = 01 \\
\chi_i^{(l)} - (\frac{1}{17} \ll 1), & \{F_4, F_5\} = 10 \\
\chi_i^{(l)}, & \{F_4, F_5\} = 11
\end{cases}
\]

As discussed in Section V-B2, the quantized slope and intercept of the linear interpolation function \(f_1\) are 0.5 and -0.125, respectively, which is convenient for hardware implementation by performing shift operations. Then \(\rho_i^{(l+1)}(\omega_{m_1})\) and \(\rho_i^{(l+1)}(\omega_{m_2})\) are calculated separately in ABS+ module as follows,

\[
\begin{align*}
\rho_i^{(l+1)}(\omega_{m_1}) &= \frac{1}{2} \cdot \tilde{\Delta}_{i}^{(l)} - \frac{1}{8} = -\frac{\tilde{\Delta}_{i,\text{imp}}^{(l)}}{2} - 0.001_2 \\
\rho_i^{(l+1)}(\omega_{m_2}) &= -\frac{1}{2} \cdot \tilde{\Delta}_{i}^{(l)} + \frac{1}{8} = \frac{\tilde{\Delta}_{i,\text{imp}}^{(l)}}{2} + 0.111_2
\end{align*}
\]

The adders used in HPA module and ABS+ module are similar to the HPA in HPA-7b+ module.
2) PIC Module: As presented in Fig. 13, PIC module consists of Mean-Sel module and MV-Mul module, which complete the calculation of $\hat{x}_l^{(i+1)}$ (Line 7 in Algorithm 3) and the calculation of $d_i^{(i+1)}$ (Line 8 in Algorithm 3), respectively.

Mean-Sel module based on Table V are illustrated in Fig. 14. Via the flags $\{F_1, F_2, F_3\}$, a two-stage selector determines selective branches of $\hat{x}_l^{(i+1)}$. In MV-Mul module, to implement the multiplication of the Gram matrix G and the vector $\hat{x}^{(i+1)} = [\hat{x}_1^{(i+1)}, ..., \hat{x}_N^{(i+1)}]^T$ with low processing latency, the matrix-vector multiplication is transformed into $2N_t$ groups of vectors’ inner-product of $g_i = [g_{i,1}, ..., g_{i,2N_t}]^T$ and $\hat{x}^{(i+1)}$, and each group is executed by HPMs and a four-stage HPA tree as shown in Fig. 15. Another adder is used to combine the output of adder tree and $\hat{d}_i$ to obtain $d_i^{(i+1)}$.

D. Time Analysis for Full-Pipeline Architecture

Table VII illustrates the time schedule of fine-granularity HF-AMP. Four iterations are fully unfolded to improve the throughput of the detector, and each iteration consists of a CPE module and a PIC module. In the non-iterative CPE module, the overall path is divided into two parts by two register groups in Fig. 12, and the processing time of each part is less than or equal to the processing time of HPM and HPA, $t_{\text{HIMA}}$, to ensure the shortest critical path of the module. In the non-iterative PIC module, the overall path is divided into four parts by four register groups as shown in Figs. 14 and 15, and each part is less than or equal to $t_{\text{HAT}}$ of the HPA tree. The main system is composed of $P = 24$-level full pipeline. Each frame is calculated within 24 clock cycles. At each moment, 24 frames are processed simultaneously, which achieves the maximum utilization of pipeline architecture.

Remark 4: For the trade-off between system throughput and area power consumption, the granularity of the pipeline needs to be determined. Different quantization schemes affect the processing delay of the computational units, which affects the critical path of the system. Therefore, we choose the maximum value $\max\{t_{\text{HIMA}}, t_{\text{HAT}}\}$ of the critical paths in the two main modules as the critical path of the system.

VI. HARDWARE IMPLEMENTATION

A. Experimental Setup

The HF-AMP in Fig. 11 is implemented in SMIC 65 nm LL 1P9M CMOS technology based on the design parameters in Section V. The HF-AMP detector is synthesized on Synopsys Design Compiler (DC) and the results are placed and routed using the Synopsys IC Compiler. The annotated toggle rate of the gate-level netlist is converted into switching activity interchange format (SAIF) for Prime-Time PX to measure the time-based chip-only power dissipation [27].

B. Implementation Results

1) ASIC Results: Denote the clock frequency as $f_{\text{clk}}$. The throughput of the HF-AMP detector can be defined as:

$$P[\text{Gb/s}] = \frac{\log_2(Q) \cdot N_t \cdot \mathcal{P}}{T_{\text{cyc}}} \cdot f_{\text{clk}},$$

where $T_{\text{cyc}}$ is the average detecting cycles.

This work presents the first hardware design of the AHPQ-assisted HF-AMP detector. Fig. 16 illustrates the layout photo, detailed summary of the chip specification, and area breakdown of the HF-AMP detector, where freq. denotes frequency. Under SMIC 65 nm CMOS technology, the whole chip consists of 537,292 k logic gates integrated within 1.208 mm² and dissipates 142.05 mW with 1.0 V supply and 360 MHz clock.

B. Implementation Results

1) ASIC Results: Denote the clock frequency as $f_{\text{clk}}$. The throughput of the HF-AMP detector can be defined as:

$$P[\text{Gb/s}] = \frac{\log_2(Q) \cdot N_t \cdot \mathcal{P}}{T_{\text{cyc}}} \cdot f_{\text{clk}},$$

where $T_{\text{cyc}}$ is the average detecting cycles.

This work presents the first hardware design of the AHPQ-assisted HF-AMP detector. Fig. 16 illustrates the layout photo, detailed summary of the chip specification, and area breakdown of the HF-AMP detector, where freq. denotes frequency. Under SMIC 65 nm CMOS technology, the whole chip consists of 537,292 k logic gates integrated within 1.208 mm² and dissipates 142.05 mW with 1.0 V supply and 360 MHz clock.

128 × 8 massive MIMO detector achieves a peak throughput of 17.92 Gb/s.

2) Comparison Between AHPQ and UQ: We implement the HF-AMP detector quantized by AHPQ and UQ separately with ASIC tools mentioned above and Xilinx Virtex-7 xc7v2000tfg 1761-1 FPGA. The results are listed in Table VIII, where syn. and eff. denote synthesis and efficiency, respectively. In the same ASIC library and environmental parameters, the chip core area
TABLE IX
ASIC IMPLEMENTATION RESULTS & COMPARISON FOR MIMO DETECTORS

| Detector        | This Work | S. Song [25] [TCASII 22] | G. Peng [23] [TCASII 18] | Y.-T. Chen [24] [TCASII’19] | W. Tang [4] [JSSC 21] | C. Foun [26] [JSSC’19] |
|-----------------|-----------|--------------------------|--------------------------|-----------------------------|----------------------|------------------------|
| Algorithm       | HF-AMP (AHPQ) | ILMPD                    | MMSE                     | Soft-MPD                    | MPD                  | LAM[1]                 |
| MIMO size       | 128 × 8   | 128 × 32                 | 128 × 8                  | 128 × 8                     | 128 × 32             |
| Modulation      | 16-QAM    | 256-QAM                  | 64-QAM                   | QPSK                        | 256-QAM              | 256-QAM               |
| Technology (SV) | SMC 65 (1.0) | TSMC 28 (0.9)            | SMIC 65 (1.0)            | TSMC 40 (0.9)               | TSMC 28 (0.9)        |                        |
| Area [mm²]      | 1.208     | 0.209                    | 2.570                    | 0.076                       | 0.580                | 0.370                 |
| Max. freq. [MHz]| 560       | 648                      | 680                      | 500                         | 425                  | 400                   |
| T/P [Gbps]      | 17.92     | 8.36                     | 1.02                     | 8.00                        | 2.76                 | 0.35                  |
| Power [mW]      | 142.05    | 19.1                     | 650.0                    | 77.93                       | 220.6                | 151.0                 |
| Energy [pJ/b]   | 7.93      | 2.28                     | 617.20                   | 9.74                        | 79.8                 | 426                   |
| Norm. T/P [Gbps]| 17.92     | 1.74                     | 1.02                     | 9.85                        | 0.45                 | 0.05                  |
| Norm. TAR[1] [Gbs/mm²] | 14.93 | 11.08                    | 0.40                     | 16.35                       | 1.66                 | 0.33                  |
| Norm. energy[2] [pJ/b] | 7.93 | 13.12                    | 617.20                   | 15.88                       | 521.13               | 2,781.20              |

1) Preprocessing unit for the matching filter b and g are not included. 2) Scaled to Nt = 8, 16-QAM and SMIC 65 nm with 1.0 V supply.

Introduction of the efficient AHPQ allows the operation nodes to be greatly simplified, ensuring that the iterative signal detection process can be fully unfolded while meeting the energy and area efficiency requirements. The minimal pipeline granularity ensures the highest throughput for our fully-unfolded pipeline architecture. Compared to MPD detector [4], we achieve an 8.99× improvement in Norm. TAR and reduce 65.72× energy consumption. Different from the MPD-based detector [25], our energy consumption of HF-AMP using UQ by about 80.1%. Meanwhile, HF-AMP using AHPQ can speed up the ASIC synthesis process by about 23.4%. For field-programmable semi-custom devices, the resource consumption of LUT slices, DSP units, and FF slices is reduced by 62.7%, 94.1%, and 74.9%, respectively.

3) Comparison to Prior Arts: To make a fair comparison, the SOA detectors are normalized to the same feature size and threshold voltage as $f_{\text{clk}} \propto \zeta, A \propto \zeta^2, E \propto \zeta$, (15) where $A$ and $E$ are core area and energy cost, respectively, and $\zeta$ and $\nu$ denote the scaling factor and voltage factor [23]. Furthermore, the system performance should be normalized to the same antenna configuration and modulation mode. Consider the number of transmitting antennas is normalized to 8, the throughput, chip area, and power consumption are all scaled by $\frac{S}{Nt \log_2 Nt}$ [5]. For the normalization of the modulation mode, the area scaling is determined by the computational complexity for soft-output MPD detectors [24]; for hard-output MPD detectors and minimum mean-squared error (MMSE) detectors, the area scaling is almost independent of the modulation mode.

Table IX summarizes the comparison results between our proposed HF-AMP using AHPQ and the SOA detectors, including ILMPD [25], MMSE [23], so-f-MPD [24], and MPD [4], where SV, max. freq., T/P, norm., and TAR mean supply voltage, maximum frequency, throughput, normalized, and throughput-to-area ratio, respectively. The proposed HF-AMP using AHPQ brings considerable improvements in both area efficiency and energy efficiency, achieving a 37.3× higher Norm. TAR and 80.4× less bit energy compared with traditional linear MMSE detector [23].

Fig. 16. (a) Layout photo, (b) chip specification, and (c) area breakdown of the HF-AMP in SMIC 65 nm technology.
implementation of AMP-based detector obtains a $1.35 \times \text{Norm.}$ TAR boost and $1.65 \times \text{Norm.}$ energy reduction. We also acquire $2.00 \times \text{energy saving of soft-MPD proposed by [24]}$ with a comparable Norm. TAR. Compared with [26], our implementation presents obvious advantages in both Norm. TAR and Norm. energy. Therefore, our proposed HF-AMP benefiting from AHPQ and fully-unfolded pipeline architecture provides superior area and energy efficiency.

VII. CONCLUSION

In this paper, an AHPQ for MIMO detectors, including PDF-based IPQ and DRL-based FPQ, is proposed. Its application on the NNA-AMP detector is analyzed, showing that AHPQ results in much lower quantization bitwidth than UQ. The hardware implementations of HF-AMP are presented to demonstrate the advantage of AHPQ. With a fully-unfolded pipeline architecture designed under SMIC 65 nm CMOS technology, the HF-AMP detector reaches a high throughput of 17.92 Gb/s with 142.05 mW for $128 \times 8$ 16-QAM MIMO. Compared to the SOA, HF-AMP using AHPQ enjoys advantages in both area and energy efficiency.

It is worth mentioning that our AHPQ scheme can also be applied to other DSP modules, not just MIMO detectors. Due to the page limit, those applications are not discussed in this paper but in our future papers.

REFERENCES

[1] T. L. Marzetta, “Noncooperative cellular wireless with unlimited numbers of base station antennas,” IEEE Trans. Wireless Commun., vol. 9, no. 11, pp. 3590–3600, Nov. 2010.
[2] Y. Peng, X. Huang, A. G. Klein, and K. Zhang, “Design and implementation of a low-complexity symbol detector for sparse channels,” IEEE Trans. Very Large Scale Integration Syst., vol. 21, no. 8, pp. 1506–1515, Aug. 2013.
[3] W. Tang, C.-H. Chen, and Z. Zhang, “A 0.58 mm$^2$ 2.76 Gb/s 79.84 pJ/b 256-QAM massive MIMO message-passing detector,” in Proc. IEEE Symp. VLSI Circuits, 2016, pp. 1–2.
[4] W. Tang, C.-H. Chen, and Z. Zhang, “A 0.58-mm$^2$ 2.76-Gb/s 79.8-pJ/b 256-QAM message-passing detector for a $128 \times 32$ massive MIMO uplink system,” IEEE J. Solid-State Circuits, vol. 56, no. 6, pp. 1722–1731, Jun. 2021.
[5] X. Tan et al., “Improving massive MIMO message passing detectors with deep neural network,” IEEE Trans. Veh. Technol., vol. 69, no. 2, pp. 1267–1280, Feb. 2020.
[6] X. Tan, W. Xu, Y. Zhang, Z. Zhang, X. You, and C. Zhang, “Efficient expectation propagation massive MIMO detector with Neumann-series approximation,” IEEE Trans. Circuits Syst. II, vol. 67, no. 10, pp. 1924–1928, Oct. 2020.
[7] X. Tan et al., “Approximate expectation propagation massive MIMO detector with weighted neumann-series,” IEEE Trans. Circuits Syst. II, vol. 68, no. 2, pp. 662–666, Feb. 2021.
[8] W. Tang, H. Prabhu, L. Liu, V. C. Wolf, and Z. Zhang, “A 1.8 Gb/s 70.6 pJ/b 128×16 link-adaptive near-optimal massive MIMO detector in 28 nm UTBB-FDSOI,” in Proc. IEEE Int. Solid-State Circuits Conf., 2018, pp. 224–226.
[9] C. Zhang, Y.-L. Ueng, C. Studer, and A. Burg, “Artificial intelligence for 5G and beyond 5G: Implementations, algorithms, and optimizations,” IEEE J. Emerg. Sel. Topics Circuits Syst., vol. 10, no. 2, pp. 149–163, Jun. 2020.
[10] W. Xu et al., “Deep learning-aided belief propagation decoder for polar codes,” IEEE J. Emerg. Sel. Topics Circuits Syst., vol. 10, no. 2, pp. 189–203, Jun. 2020.
[11] Y.-S. Jeon, M. So, and N. Lee, “Reinforcement-learning-aided ML detector for uplink massive MIMO systems with low-precision ADCs,” in Proc. IEEE Wireless Commun. Netw. Conf., 2018, pp. 1–6.

Yingmeng Ge received the B.S. degree in information engineering from Southeast University, Nanjing, China, in 2018. He is currently working toward the Ph.D. degree in information science and engineering with the National Mobile Communications Research Laboratory, Southeast University. His research interests include MIMO detection and AI in communications.

Zhenhao Ji received the B.S. degree in information engineering from Southeast University, Nanjing, China, in 2021. He is currently working toward the M.S. degree in communication system with the National Mobile Communications Research Laboratory, Southeast University. He was the World Winner of 2019–2020 IEEE CASS Student Design Competition. His research interests includes AI in communication and VLSI design for digital signal processing.
Yongming Huang (Senior Member, IEEE) received the B.S. and M.S. degrees from Nanjing University, Nanjing, China, in 2000 and 2003, respectively, and the Ph.D. degree in electrical engineering from Southeast University, Nanjing, in 2007. Since March 2007, he has been a Faculty with the School of Information Science and Engineering, Southeast University, where he is currently a Full Professor. During 2008–2009, he visited the Signal Processing Lab, Royal Institute of Technology, Stockholm, Sweden. He has authored or coauthored more than 280 papers and issued 90 patents. His research interests include intelligent 5G/6G mobile communications and millimeter wave wireless communications. He was the recipient of certificate of appreciation for outstanding contribution to the development of IEEE standard 802.11aj. He was an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and the Guest Editor of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS. He is currently the Editor-at-Large of the IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY and an Associate Editor for the IEEE WIRELESS COMMUNICATIONS LETTERS.

Zaichen Zhang (Senior Member, IEEE) was born in Nanjing, China, in 1975. He received the B.S. and M.S. degrees in electrical and information engineering from Southeast University, Nanjing, in 1996 and 1999, respectively, and the Ph.D. degree in electrical and electronic engineering from The University of Hong Kong, Hong Kong, China, in 2002. From 2002 to 2004, he was a Postdoctoral Fellow with the National Mobile Communications Research Lab., Southeast University. From 2004, he joined the Department of Information Science and Engineering, Southeast University where he is currently a Professor. He has authored or coauthored more than 280 papers and issued 90 patents. His research interests include 6G wireless communications, optical wireless communications, and quantum information.

Chuan Zhang (Senior Member, IEEE) received the B.E. degree in microelectronics and the M.E. degree in very-large scale integration design from Nanjing University, Nanjing, China, in 2006 and 2009, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA, in 2012. He is currently the Young Chair Professor with Southeast University, Nanjing. He is also with the LEADS, National Mobile Communications Research Laboratory, Frontiers Science Center for Mobile Information Communications and Security of MoE, Quantum Information Center of Southeast University, and the Purple Mountain Laboratories, Nanjing. His research interests include algorithms and implementations for signal processing and communication systems. Dr. Zhang is an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS - II, IEEE TRANSACTIONS ON SIGNAL PROCESSING, and IEEE OPEN JOURNAL OF CIRCUITS AND SYSTEMS, and the Corresponding Guest Editor of the IEEE JOURNAL ON EMERGING AND SELECTED TOPICS IN CIRCUITS AND SYSTEMS twice. He is a Distinguished Lecturer and the Vice Chair of the Circuits and Systems for Communications TC of the IEEE Circuits and Systems Society. He is also a Member of the Applied Signal Processing Systems TC of the IEEE Signal Processing Society, and Circuits and Systems for Communications TC, VLSI Systems and Applications TC, and Digital Signal Processing TC of the IEEE Circuits and Systems Society. He was the recipient of the Best Contribution Award of the IEEE Asia Pacific Conference on Circuits and Systems (APCCAS) in 2018, Best Paper Award in 2016, Best (Student) Paper Award of the IEEE International Conference on DSP in 2016, three Best (Student) Paper Awards of the IEEE International Conference on ASIC in 2015, 2017, and 2019, Best Paper Award Nominations of the IEEE Workshop on Signal Processing Systems in 2015, three Excellent Paper Awards and two Excellent Poster Presentation Awards of the International Collaboration Symposium on Information Production and Systems from 2016 to 2018, Outstanding Achievement Award of the Intel Collaborative Research Institute in 2018, and Merit (Student) Paper Award of the IEEE APCCAS in 2008. He also received the Three-Year University-Wide Graduate School Fellowship of UMN and Doctoral Dissertation Fellowship of UMN.

Xiaohu You (Fellow, IEEE) received the M.S. and Ph.D. degrees in electrical engineering from Southeast University, Nanjing, China, in 1985 and 1988, respectively. Since 1990, he has been with the National Mobile Communications Research Laboratory, Southeast University, where he is currently the Director and a full Professor. He has long been engaged in teaching, scientific research, and organizational management of mobile communication systems, and made important contributions to the development of China’s 3G, 4G, and 5G mobile communications. From 1999 to 2002, he was a Principal Expert of the C3G Project, responsible for organizing China’s 3G Mobile Communications Research and Development Activities. From 2001 to 2006, he was a Principal Expert of the China National 863 Beyond 3G FuTURE Project. Since 2013, he has been a Principal Investigator of the China National 863 Project. He has contributed more than 200 IEEE journal articles and two books in the areas of adaptive signal processing and neural networks, and their applications to communication systems. His research interests include mobile communication systems, and signal processing and its applications. He was the recipient of the 1st Class National Technological Invention Award, 2nd Class National Science and Technology Progress Award, Tan Kah Kee Science Award, and other important awards. He was Elected as the IEEE Fellow for his contributions to the development of mobile communications in China in 2011. He was the General Chair of the IEEE Wireless Communications and Networking Conference 2013, IEEE Vehicular Technology Conference 2016, and IEEE International Conference on Communications 2019. He is the Secretary-General of the FuTURE Forum and Vice-Chair of the China IMT-2020 Promotion Group and China National Mega Project on New Generation Mobile Network.