Approximate Floating-Point FFT Design with Wide Precision-Range and High Energy Efficiency

Chenyi Wen, Ying Wu, Xunzhao Yin, Cheng Zhuo

Zhejiang University, Hangzhou, China

Jun. 17, 2023

Biography

Chenyi Wen

Zhejiang University
wwency@zju.edu.cn

Chenyi Wen is a Ph.D. student supervised by Prof. Cheng Zhuo at College of Information Science and Electronic Engineering, Zhejiang University. She received the B.S. degree in Microelectronics Science and Engineering from Zhejiang University in 2022. Her research interests include approximate computing and low power optimization.

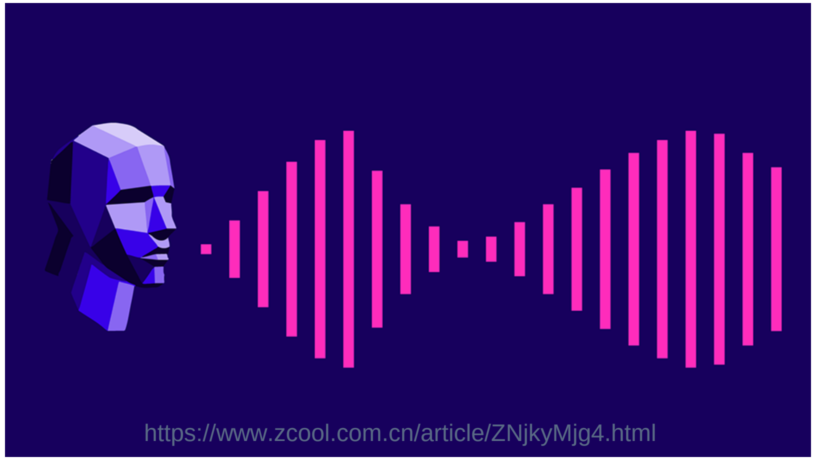
Outline

- Background
- Overview
- Method
- Results
- Conclusions

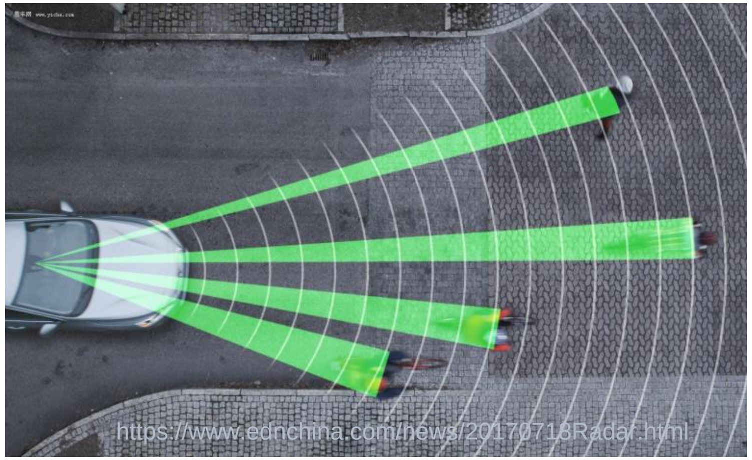
Background - Fast Fourier Transform (FFT)

- **Fast Fourier Transform (FFT)** is a **fundamental algorithm** in digital communication and signal processing

Image Processing



Speech Recognition



Radar Positioning

- For **FFT accelerators** in resource-constrained systems, they need to achieve **both high performance and energy efficiency**

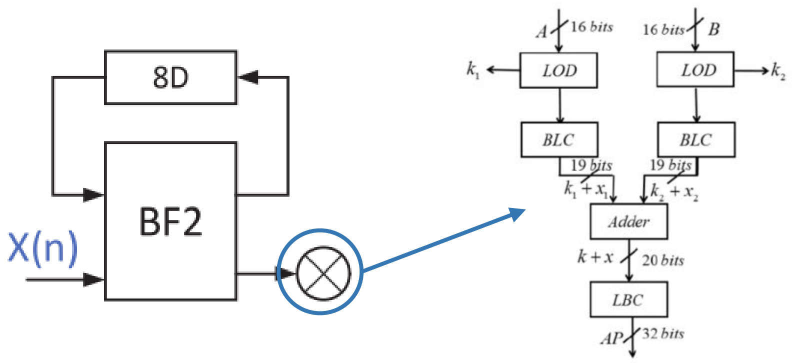
Background - Approximate FFT Accelerator

- The requirements of full precision and exactness are **not always necessary** for FFT operations

- Explore approximate design of FFT to achieve **sufficient** instead of excessively accurate computational precision

Background - Related Work

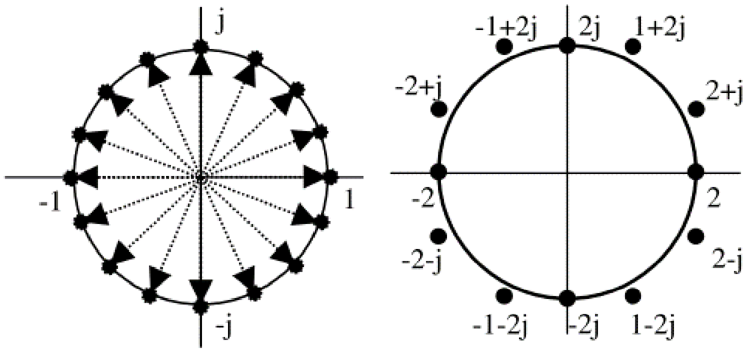
Circuit Level Approximation



- **Method**: directly approximate the underlying circuits to replace exact units
- **Limitation**: **ineffective optimization** due to **missing link** between FFT algorithm precision and introduced approximation

[A. K. Y. Reddy, et al., IEEE ICCES, 2018]

Algorithm Level Approximation



- **Method**: fine-tune the value of the rotation factor to reduce multiplication complexity
- **Limitation**: **low flexibility** to support versatile applications

[V. Ariyarathna, et al., IEEE TAP, 2019]

Background - Challenges

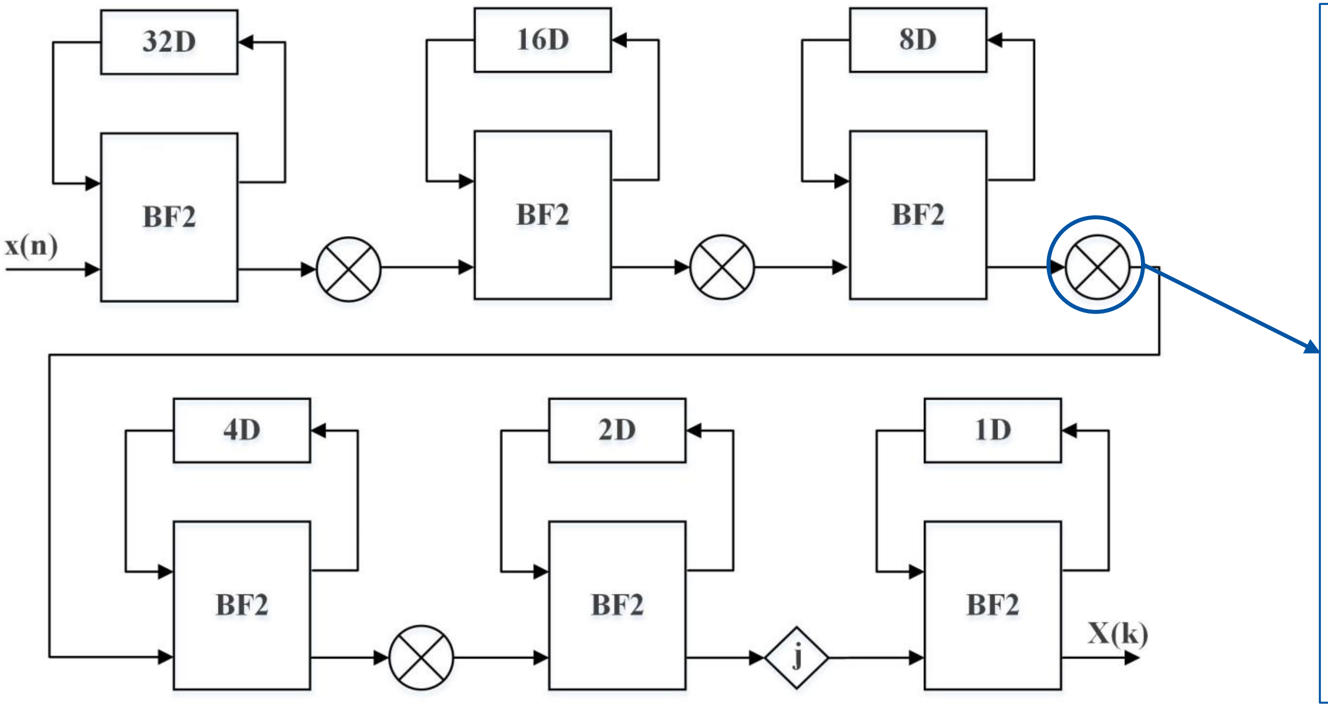
- **Ineffective optimization** due to **missing link** between the FFT algorithm precision and the introduced approximation
  - **Explore the relationship** between the introduced circuit level approximations and the algorithm level precision requirements
  - **Optimize the designed FFT** to maximize the benefits of approximate computing
- **Low flexibility** to support versatile applications
  - **Design configurable circuit** to support different applications

Outline

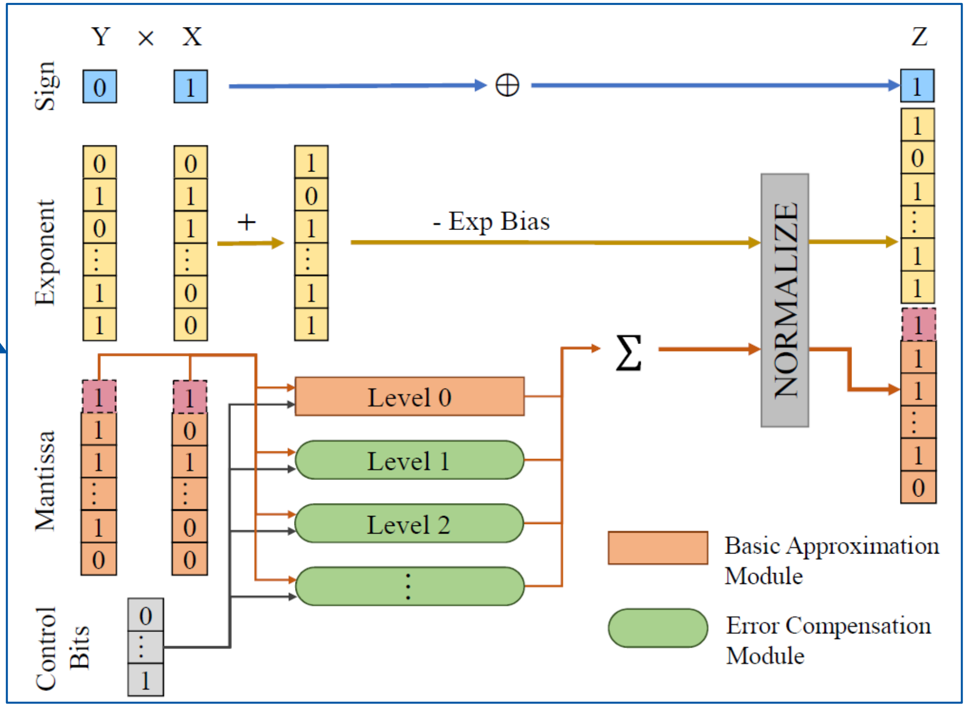
- Background
- Overview
- Method
- Results
- Conclusions

Overview - Baseline of Our Work

- R2-SDF **pipeline FFT** based on DIF algorithm



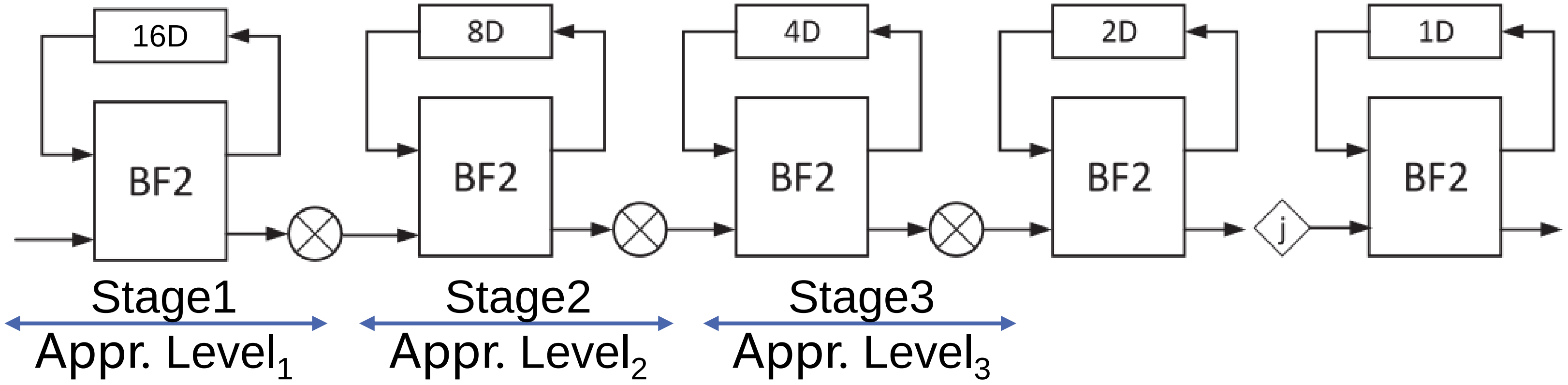
- Piecewise-linearly-**Approximated** floating-point **Multiplier** (PAM)



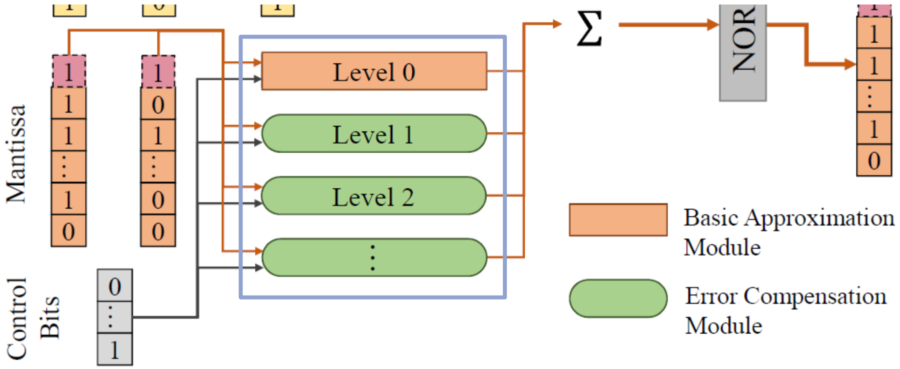
[Chuangtao Chen, et al., ICCAD, 2020]

Overview - A Top-down Approximate FFT Design Methodology

Optimization | Exploit the error-tolerance nature with the FFT precision specification



Circuit | Multi-Lever & Configurable



Analysis | Link circuits to algorithm accuracy

- **Circuit error** introduced under approximate levels
- **FFT algorithm precision** impacted by the circuit error

Overview - Methodology Details

Error Modeling

Error Characteristics Analysis

↓

Error Model Construction

↓

FFT Precision Calculation

→ Error Model →

Approximation Optimization

- Satisfy accuracy specification
- Minimize hardware overhead

$$\textbf{Minimize} \quad \sum n_{stage}$$

$$\textbf{Subject to}: \quad PSNR_{bnd} > PSNR_{spec}$$

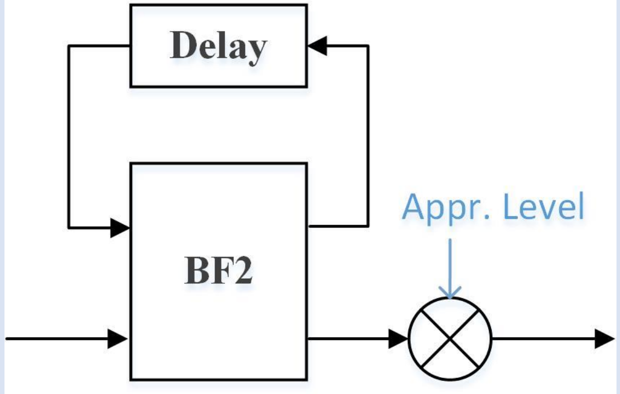
$$P(x \mid x \in Set_m, x \geq PSNR_{bnd}) > Prob_{spec}$$

$$n_{stage} \text{ is integer}, n_{stage} \in [0,11]$$

→ Appr. Level →

Design Implementation

- Approximation Error Tolerance
- Approximation Balance



Outline

Error Modeling - Error Characteristics Analysis

- **Calculate the exact error**
  - An FP number: $F_x = sign_x \times x \times 2^{E_x}$
  - Error introduced by the approximate multiplier PAM:

$$error^n = -sign_x sign_y \times (x - k_x)(y - k_y) \times 2^{E_x + E_y}$$

$$k_x = [0.5 + floor(x \times 2^n)] \times 2^{-n}$$
$$k_y = [0.5 + floor(y \times 2^n)] \times 2^{-n}$$

Depend on F_x, F_y and approximate level n

- **Find a conservative error measure**
  - If FP numbers $F_x, F_y \in [-bnd, bnd]$ and bnd happens to be the power of 2

$$error^n \leq |error^n_{bnd}| = 4^{E_{bnd} - n - 1}$$

Depend on bnd and n

How to make $error^n$ only depend on the approximate level n?

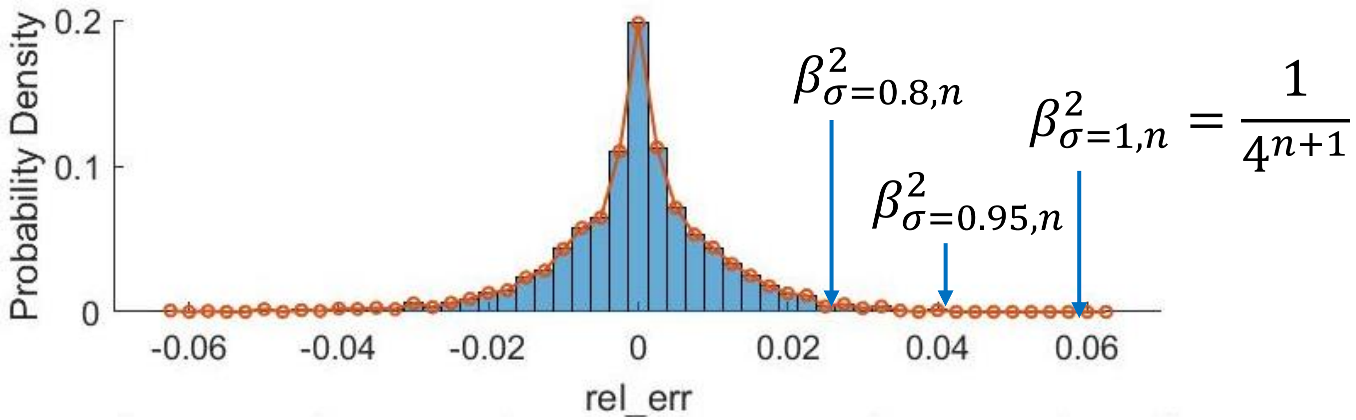
Error Modeling - Error Model Construction

- **Eliminate all the parameters except the approximate level *n***
  - ▸ The relative error for the product of $F_x F_y$:

$$rel\_err^n(F_x, F_y) = \frac{error^n}{F_x F_y} = \frac{-(x-k_x)(y-k_y)}{xy} \leq \frac{1}{4^{n+1}}$$

Only depend on n

- **Use the error at a particular percentile $\beta^2_{\sigma,n}$ for estimation**
  - ▸ The distribution of rel_err^n with x, y uniformly selected in [1,2):

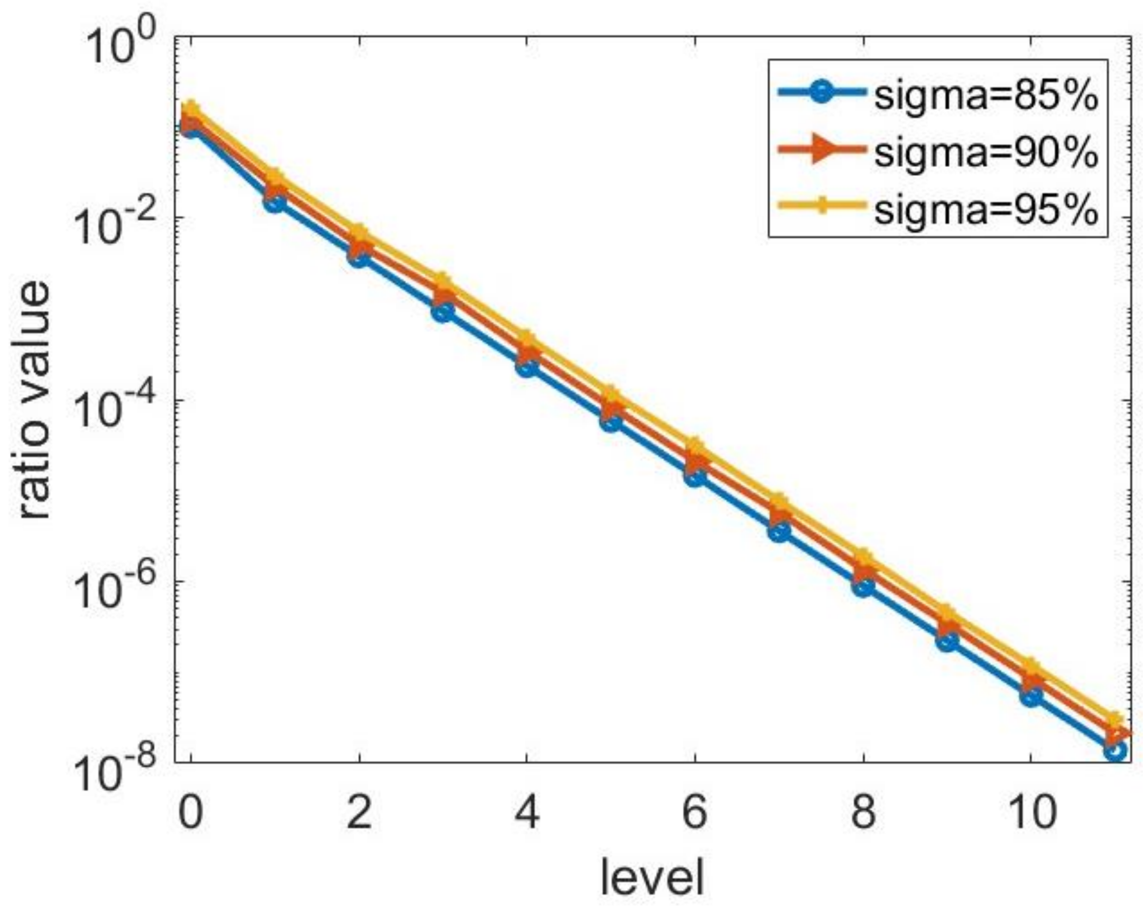


$$F_x F_y + error^n$$
$$= F_x F_y(1 + rel\_err^n(F_x, F_y))$$
$$\approx F_x F_y(1 + \beta^2_{\sigma,n})$$

Why we replace $error^n$ with $\beta^2_{\sigma,n}$?

Error Modeling - Error Model Construction

- **Why we replace $error^n$ with $\beta_{\sigma,n}^2$**
  - $\beta_{\sigma,n}^2$ can be **pre-characterized** with certain n and σ, while $error^n$ cannot
  - $\beta_{\sigma,n}^2$ **effectively simplifies** the error estimation, while $error^n$ is complex
  - Sometimes it is hard to calculate the exact $error^n$, while $\beta_{\sigma,n}^2$ is much **easier to calculate**



Error Modeling - FFT Precision Calculation

- A 64-point R2DIF algorithm is used as a demonstration example:

$$O_s[i] = (D_{s-1}[i] \pm D_{s-1}[i \pm 2^{6-(s-1)}] + \theta_{s-1}^2)W_s[i] + \theta_s^2$$

θ^2: the introduced approximate error
W: the rotation factor which is always less than 1

  - θ_{s-1}^2 may get reduced by W and hence we can assign larger approximations in prior stages → **Approximation error tolerance**
  - If θ_{s-1}^2 is a large error and dominates the entire FFT, it will be meaningless to reduce θ_s^2 for improving the overall accuracy
    → **Approximation balance**

Approximation Optimization

- Formulate the problem as a Mixed-Integer Nonlinear Optimization Problem (MINLP):

$$\boxed{\textbf{Minimize} \sum n_{stage}}$$

→ Maximize the possible approximation

$$\boxed{\begin{aligned}\textbf{Subject to: } & PSNR_{bnd} > PSNR_{spec} \\ & P(x \mid x \in Set_m,\ x \geq PSNR_{bnd}) > Prob_{spec} \\ & n_{stage} \in [0, 11]\end{aligned}}$$

↓

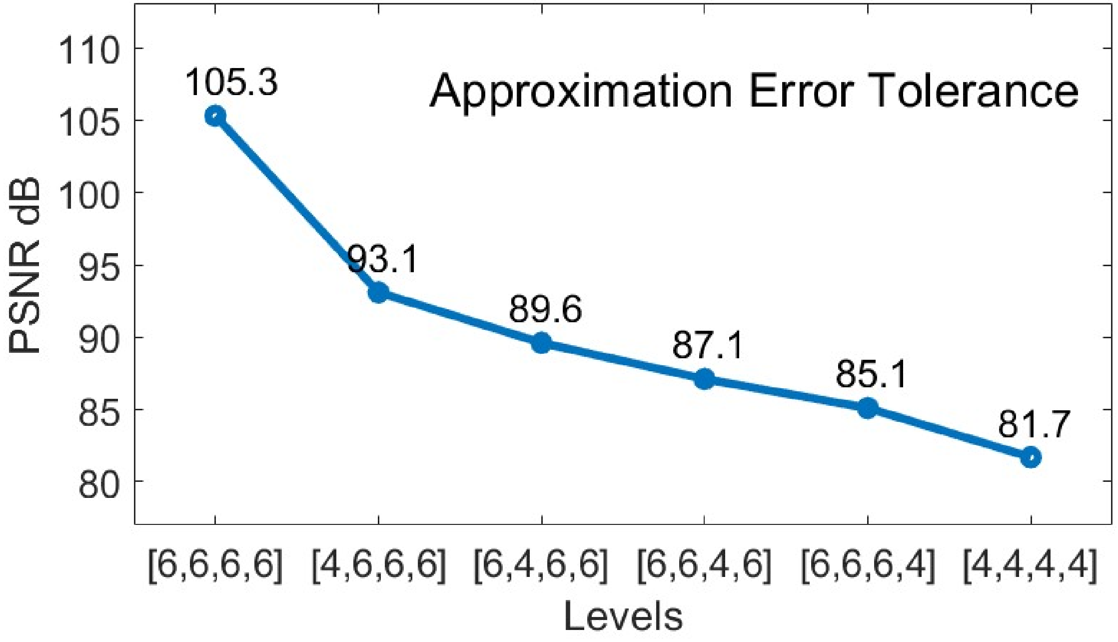
Ensure the algorithm accuracy of the optimized design to satisfy the precision specification

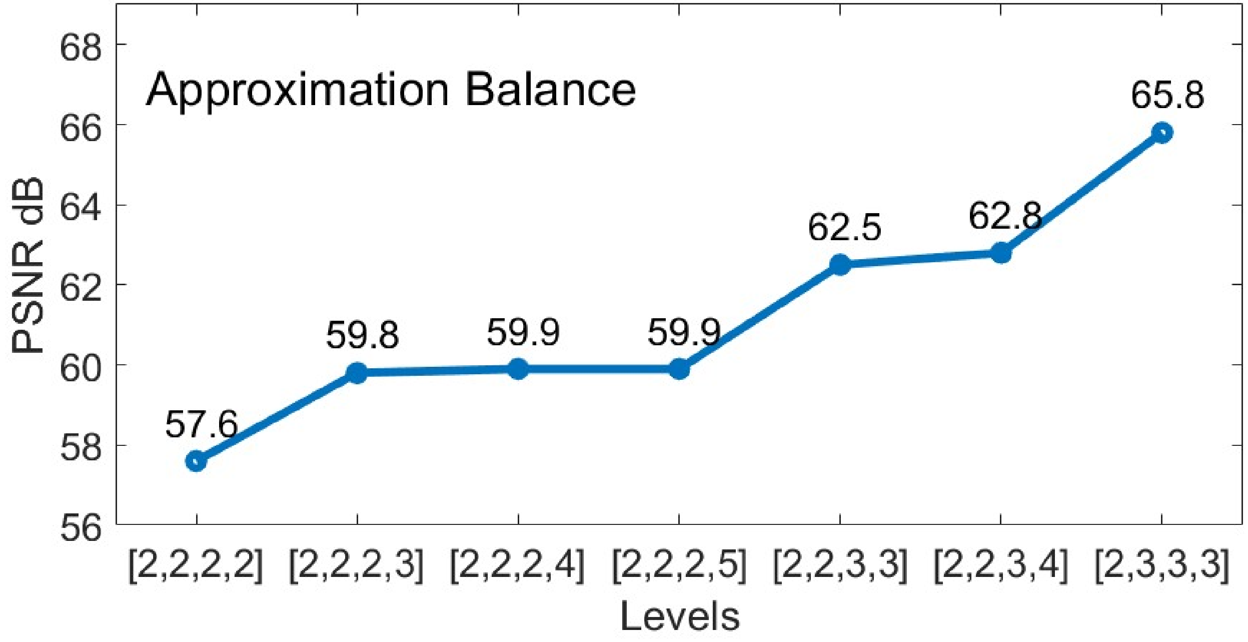
Design Implementation

- Two principles to guide the design implementation:
  - **Approximation error tolerance**

    PSNR decreases less if the earlier stage is assigned with a low approximate level
  - **Approximation balance**

    The accuracy increase in the last stage does not help the overall accuracy



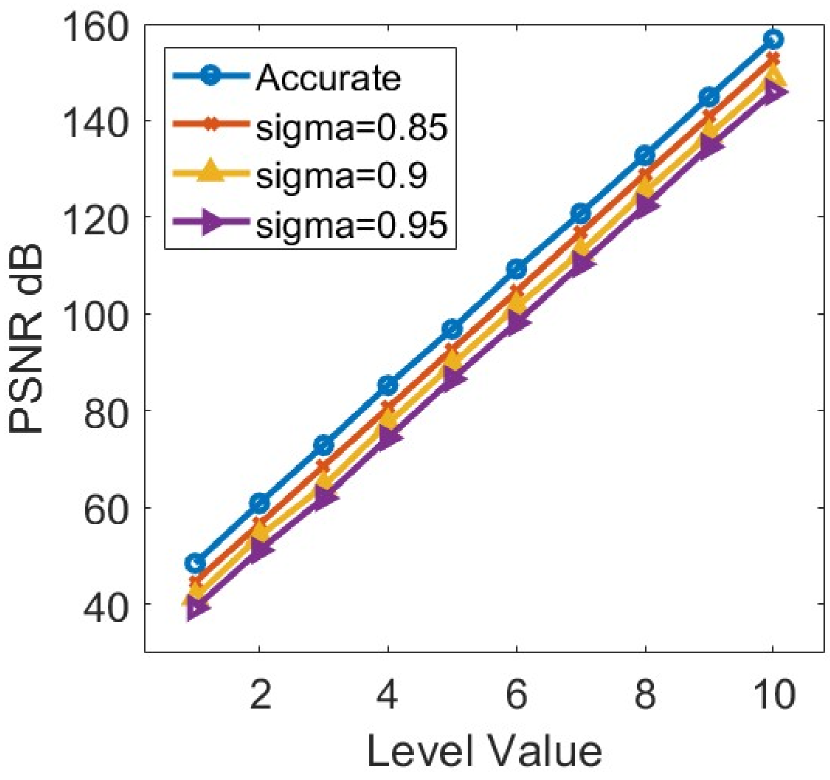


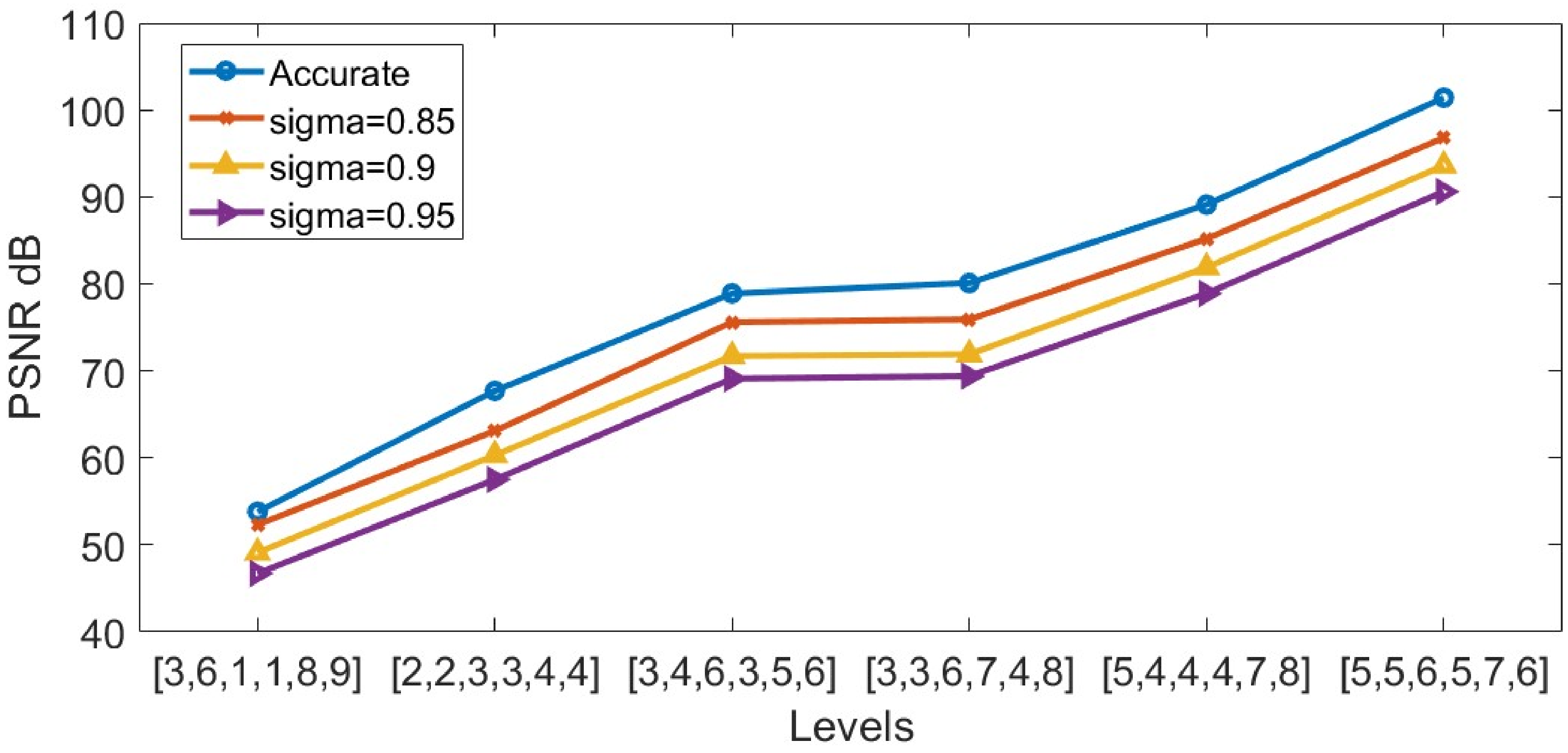
Outline

- Background
- Overview
- Method
- Results
- Conclusions

Performance of Error Model

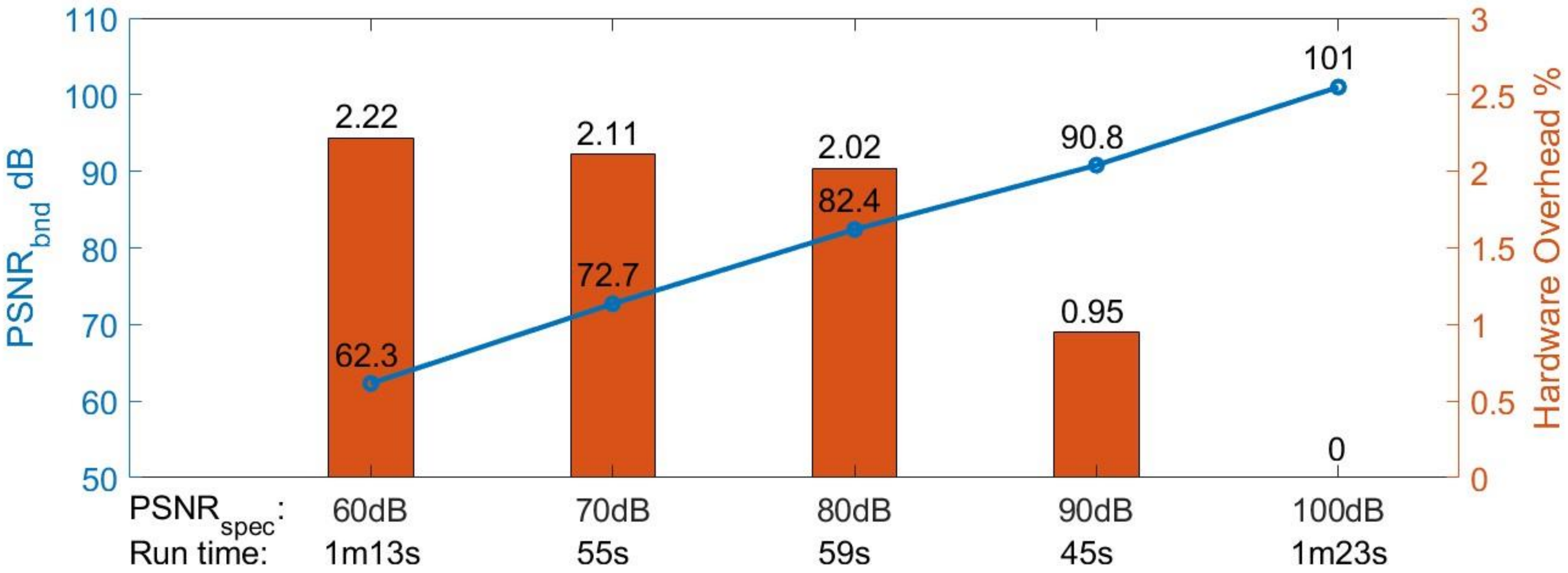
- Validate the accuracy of the proposed error model on a 256-point R2DIF FFT
  - **Small deviations** to accurate PSNR
  - **Controllable conservativeness** with an optional σ





Performance of Optimization

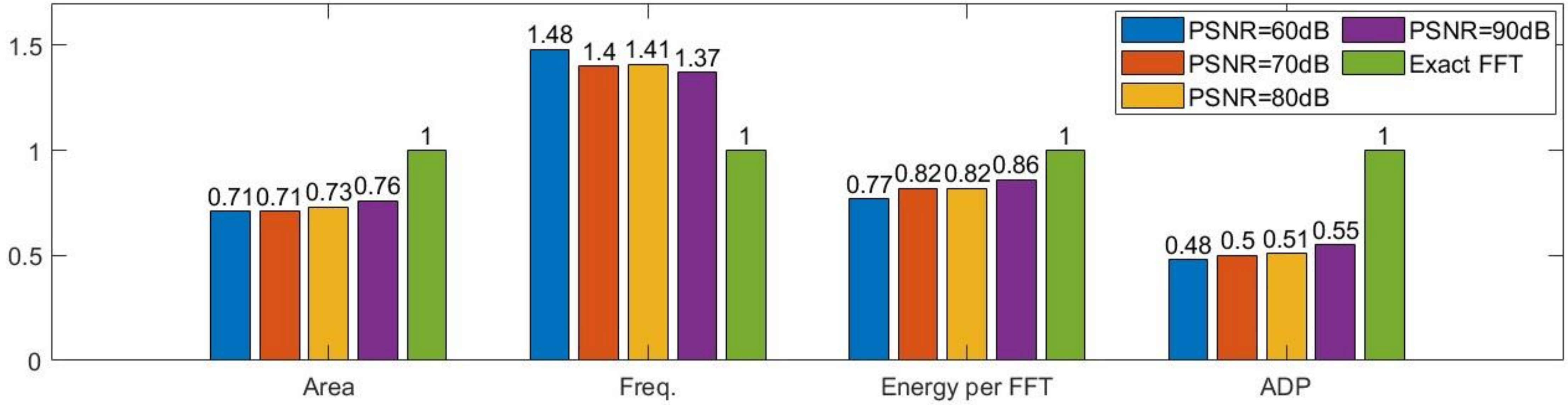
- Validate the proposed optimization flow with different PSNR specifications for a 256-point R2-DIF FP FFT
  - Solutions are **close to the optimal** combinations
  - Significantly **decrease design time**



Approximate FFT Design Comparison

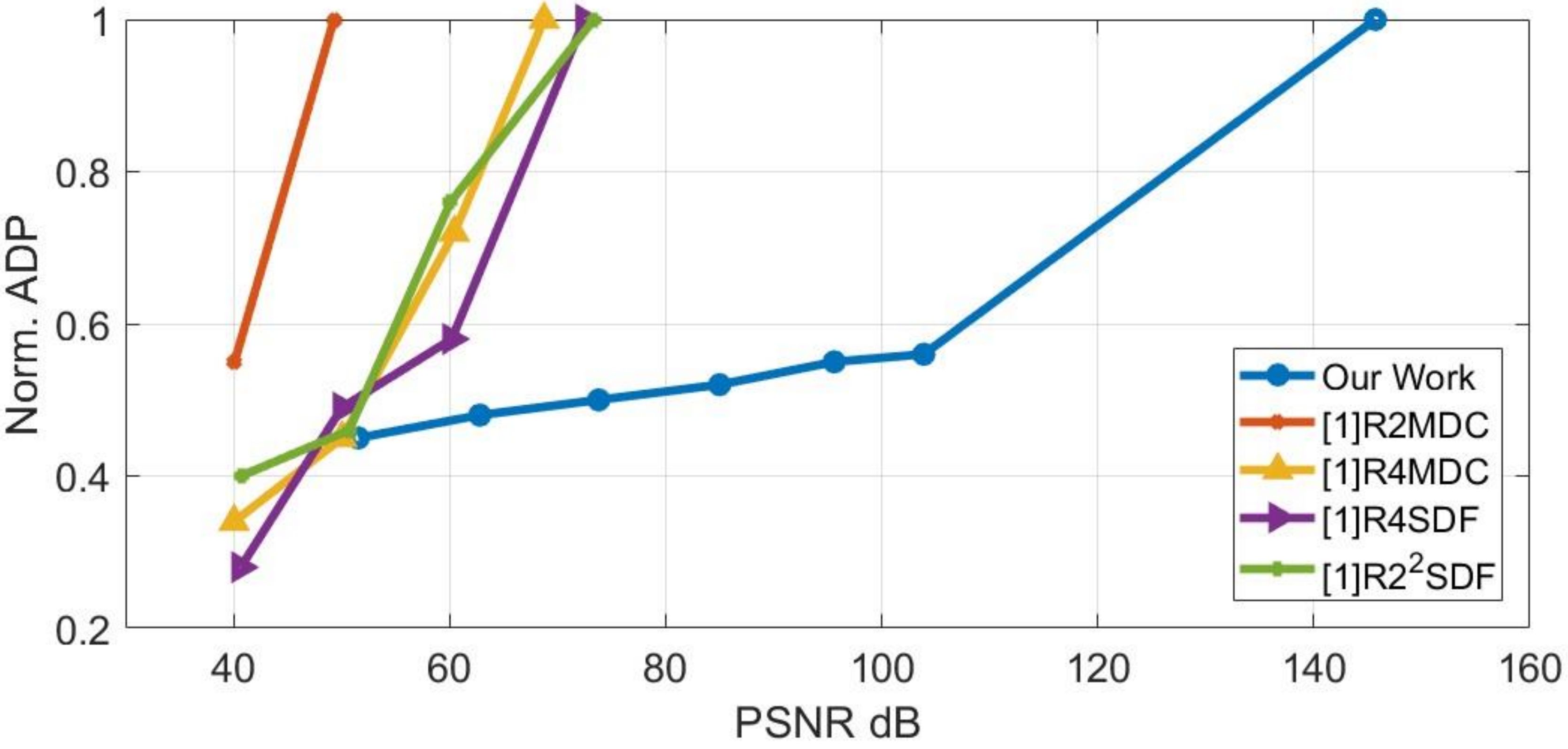
- Comparison between the approximate FFT and an exact FFT
  - Satisfy the PSNR requirements with smaller than 6.5% difference
  - **20%** area saving
  - **40%** speed improvement
  - **15%** energy saving

No.	1	2	3	4	5
$PSNR_{spec}$ (dB)	60	70	80	90	Exact
$PSNR_{act}$ (dB)	62.8	73.8	85	95.6	145.7
Deviation	4.67%	5.42%	5.88%	6.22%	-



Approximate FFT Design Comparison

- Comparison between the approximate FFT and prior state-of-the-art approximate FFT designs
  - **Wider precision range**
  - **Higher energy efficiency** with tight PSNR constraint



[1] [Weiqiang Liu, et al., IEEE Transactions on Circuits and Systems, 2019]

Conclusions

- **A top-down approximate FFT design methodology**
  - Fully exploit the error-tolerance nature of the FFT algorithm
  - Automatically determine the appropriate approximate levels
- An FFT approximation **error model**
  - Bound the impact of circuit approximation on the FFT algorithm precision
- An FFT approximation **optimization flow**
  - Maximize the energy efficiency while meeting the design specifications
- Achieve almost **2× wider precision-range** and **higher energy-efficiency** when compared to the prior approximate FFT designs

THANK YOU!