MONTE CARLO RESPONSE-TIME ANALYSIS

RTSS 2021
9 December 2021

Sergey Bozhko, Georg von der Brüggen*, and Björn Brandenburg
* now at TU Dortmund, Germany

MAX PLANCK INSTITUTE FOR SOFTWARE SYSTEMS

European Research Council
Established by the European Commission

MAX-PLANCK-GESELLSCHAFT

MAIN CONTRIBUTIONS

A new application of Monte Carlo technique in RTS

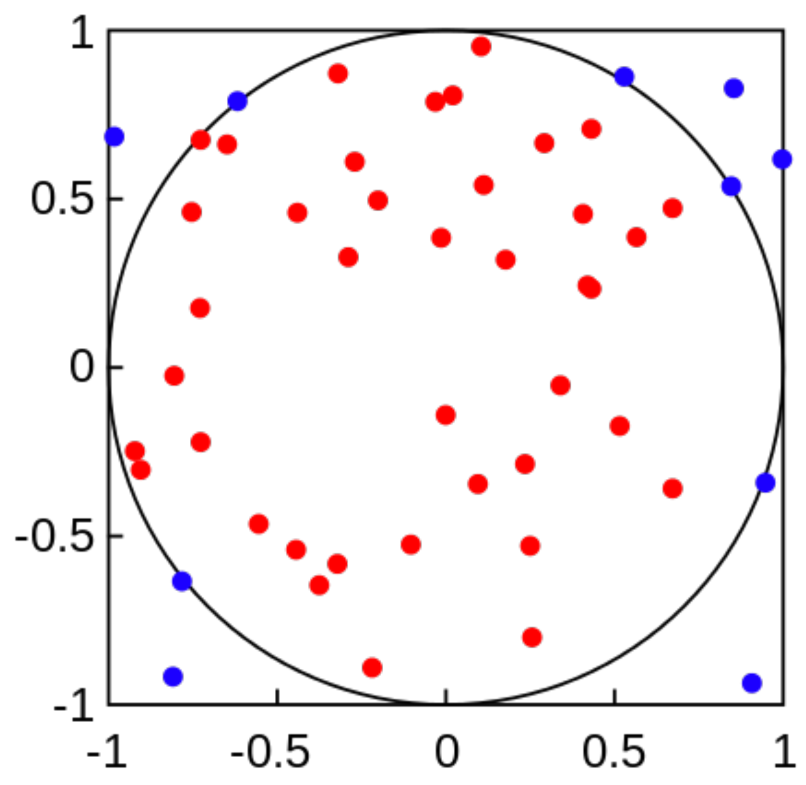
➔ The first paper that applies Monte Carlo to probabilistic response-time analysis

A new algorithm for Worst-Case Deadline Failure Probability estimation

➔ Less sensitive to input parameters than state-of-the-art

➔ Easy parameter tuning

➔ In most cases outperforms state-of-the-art approaches



https://en.wikipedia.org/wiki/Monte_Carlo_integration

A CASE FOR PROBABILISTIC RTA

SURVEY OF INDUSTRY PRACTICE IN RTS

Soft real-time systems are quite popular! [Akesson et. al, 2020]

➔ 62% of respondents: system includes soft or firm real-time components

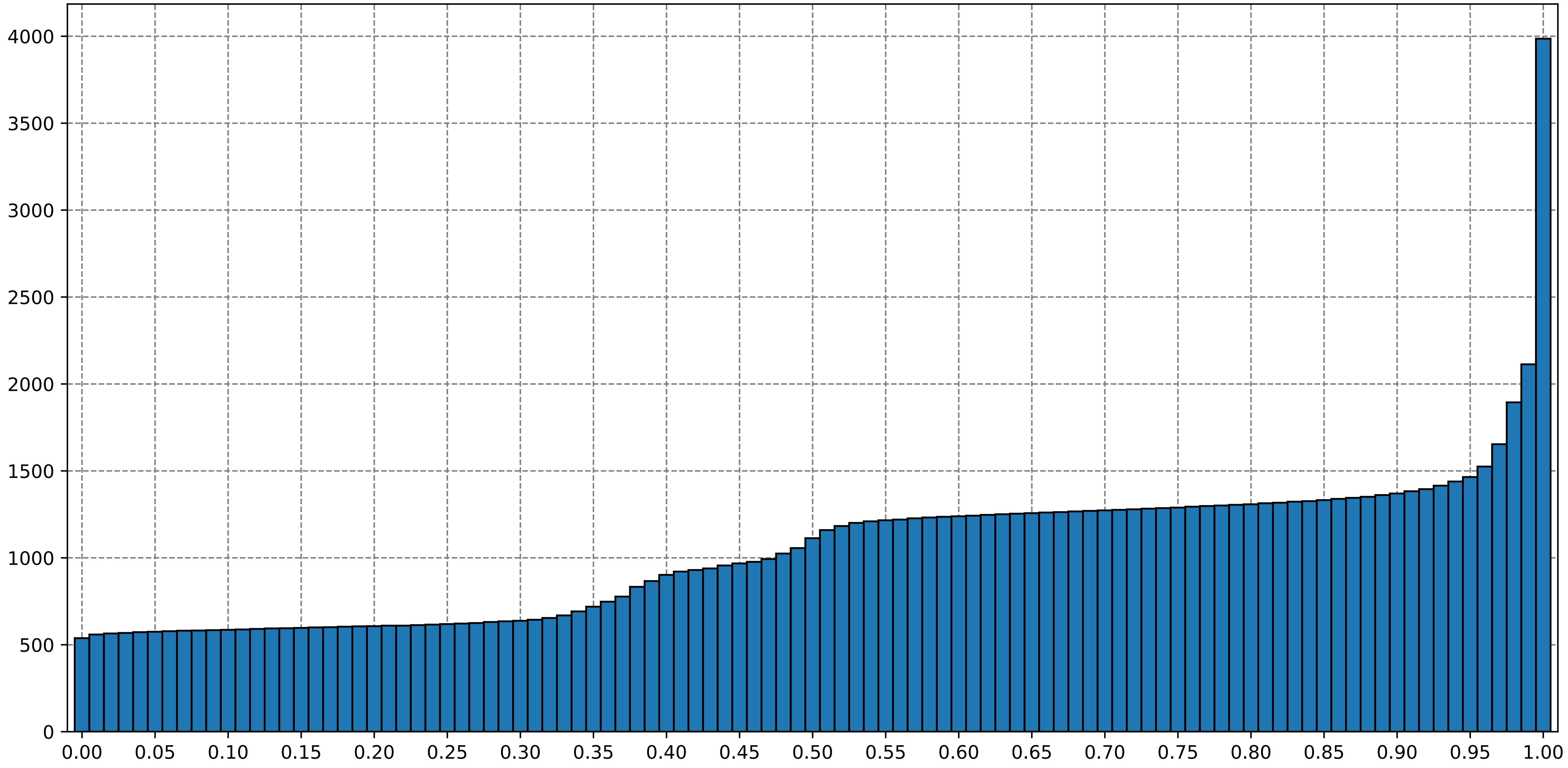
➔ 45% of respondents: the most critical function can miss some deadlines

True hard real-time systems are rare

➔ (Only) 15% of respondents: deadlines can never be missed

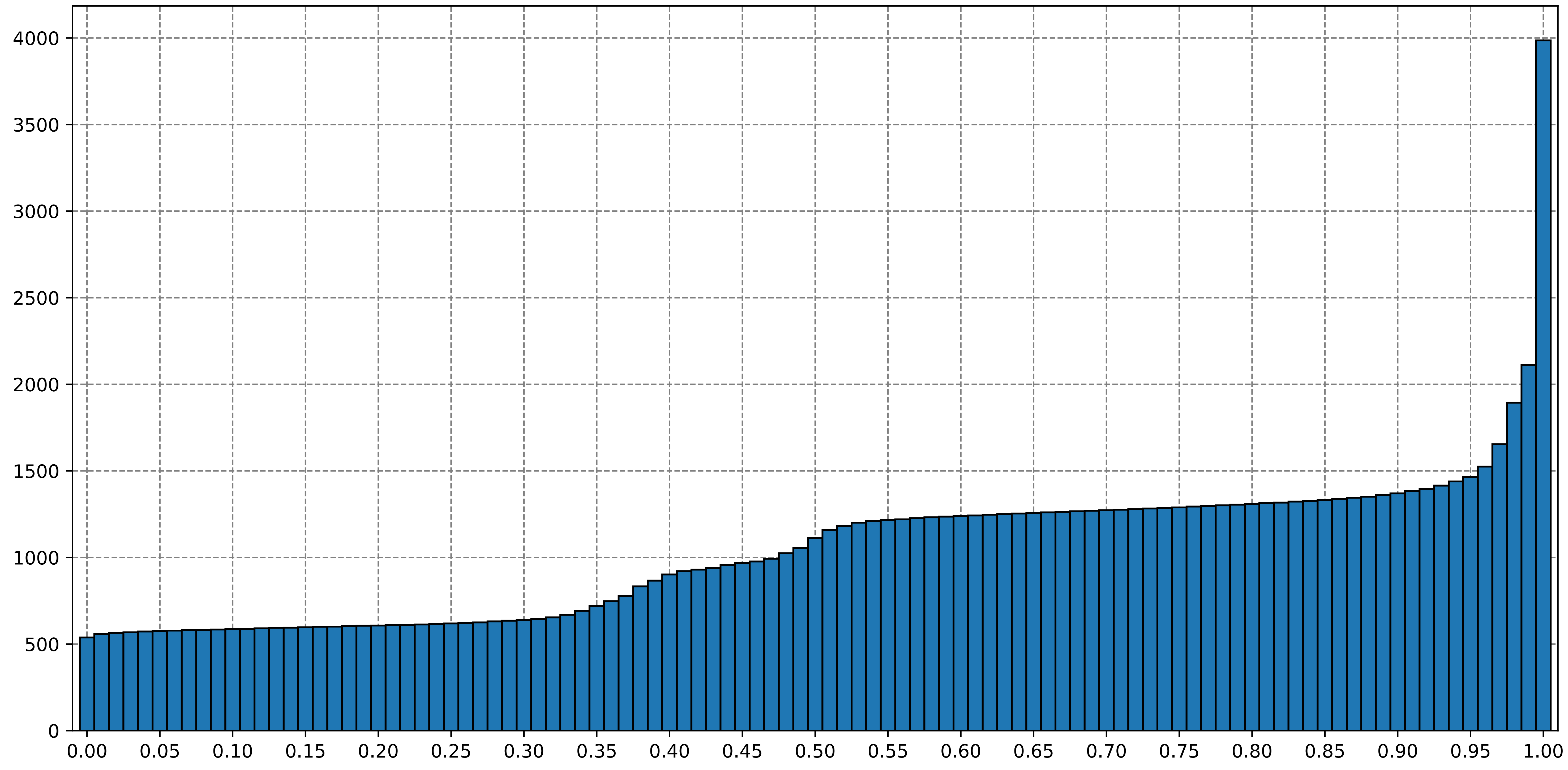
B. Akesson, M. Nasri, G. Nelissen, S. Altmeyer, and R. I. Davis, "A comprehensive survey of industry practice in real-time systems", RTSS 2020

THE NEED FOR BELOW-WORST-CASE PROVISIONING



B. Brandenburg and M. Gül, "Global scheduling not required: Simple, near-optimal multiprocessor real-time scheduling with semi-partitioned reservations", RTSS 2016

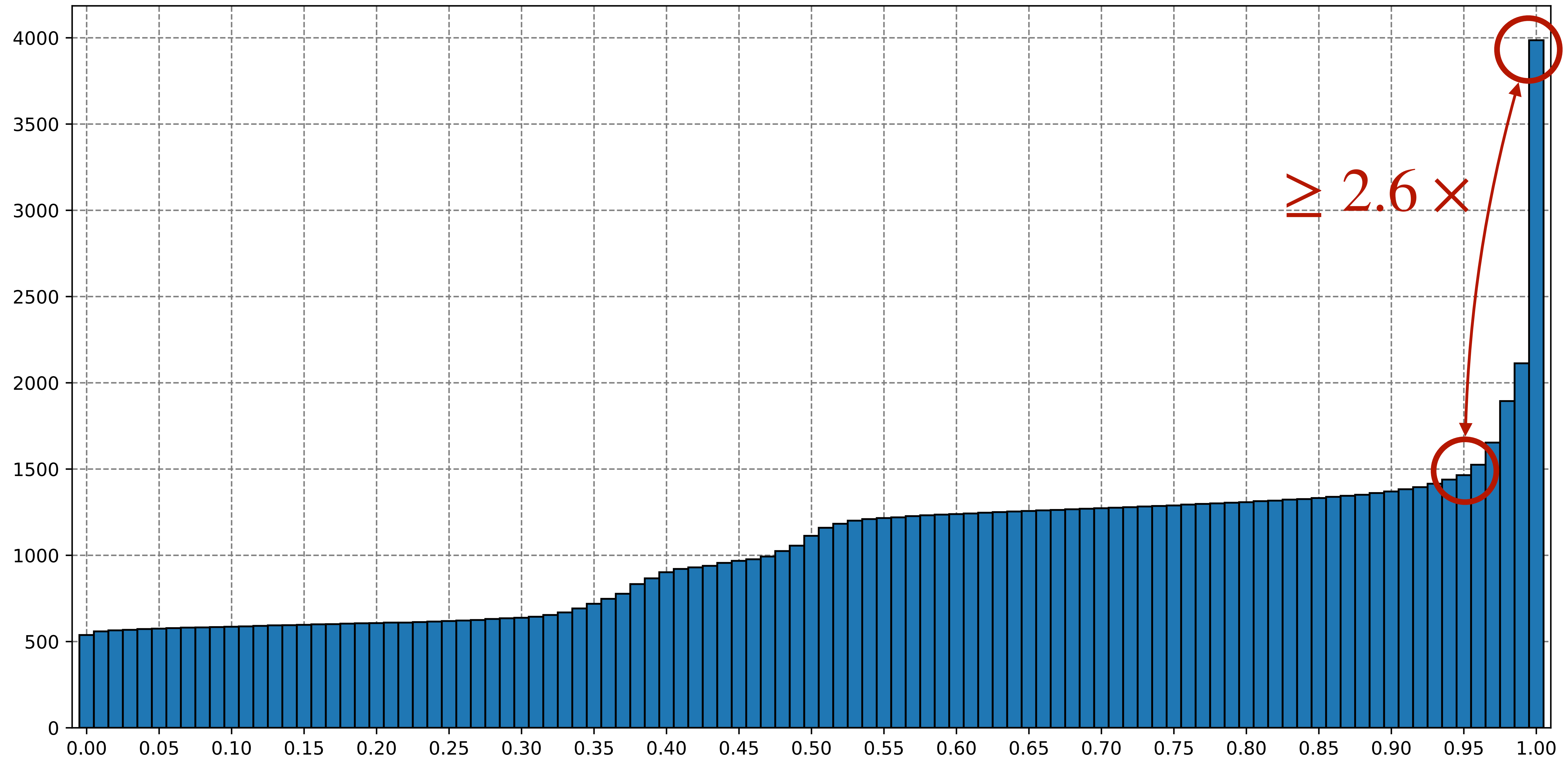
THE NEED FOR BELOW-WORST-CASE PROVISIONING



WCET Setting: $\tau_i = (C_i = 4000,\ T_i = 5000,\ D_i = 5000)$

B. Brandenburg and M. Gül, "Global scheduling not required: Simple, near-optimal multiprocessor real-time scheduling with semi-partitioned reservations", RTSS 2016

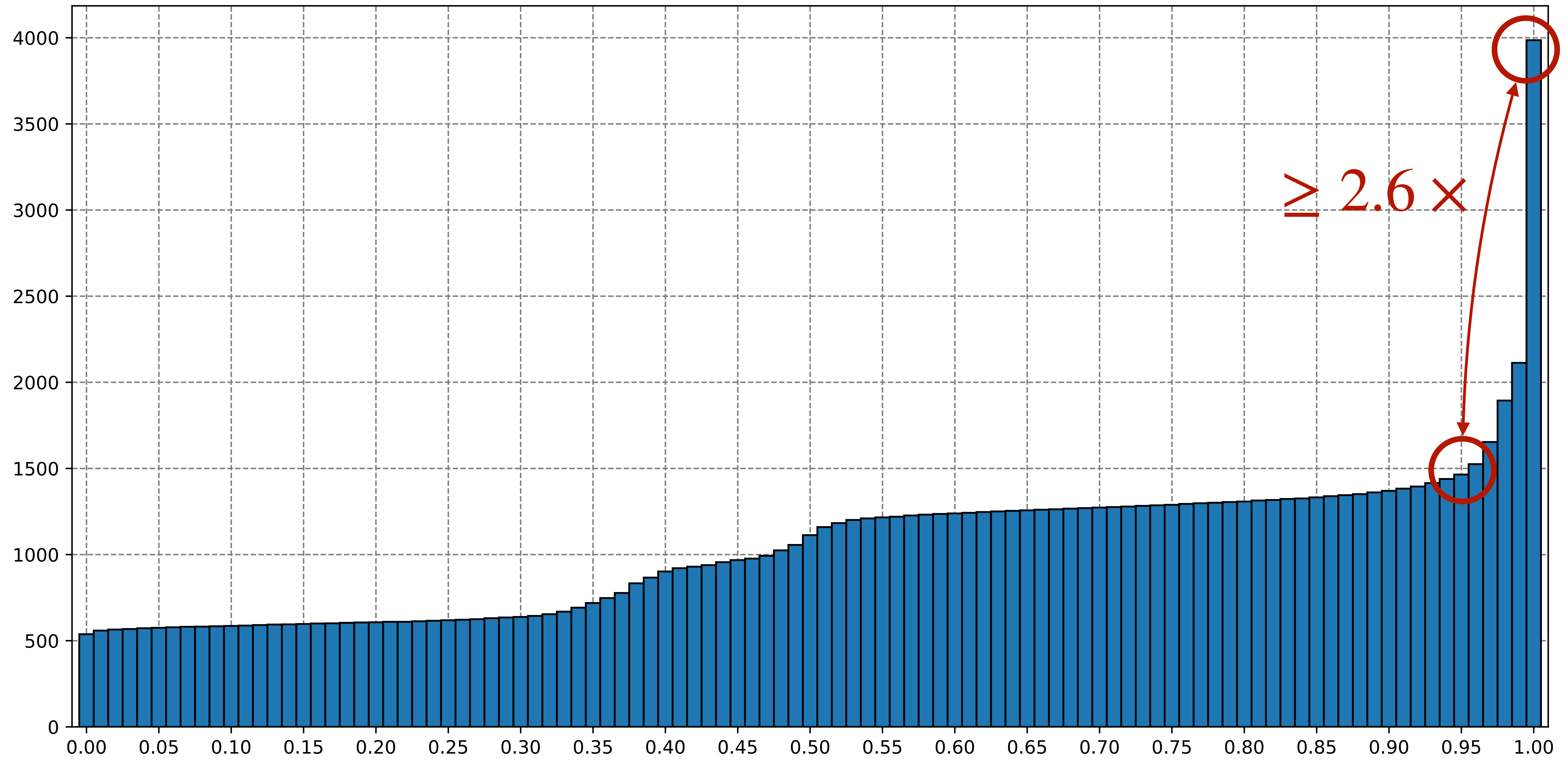
THE NEED FOR BELOW-WORST-CASE PROVISIONING



WCET Setting: $\tau_i = (C_i = 4000,\ T_i = 5000,\ D_i = 5000)$

B. Brandenburg and M. Gül, "Global scheduling not required: Simple, near-optimal multiprocessor real-time scheduling with semi-partitioned reservations", RTSS 2016

THE NEED FOR BELOW-WORST-CASE PROVISIONING

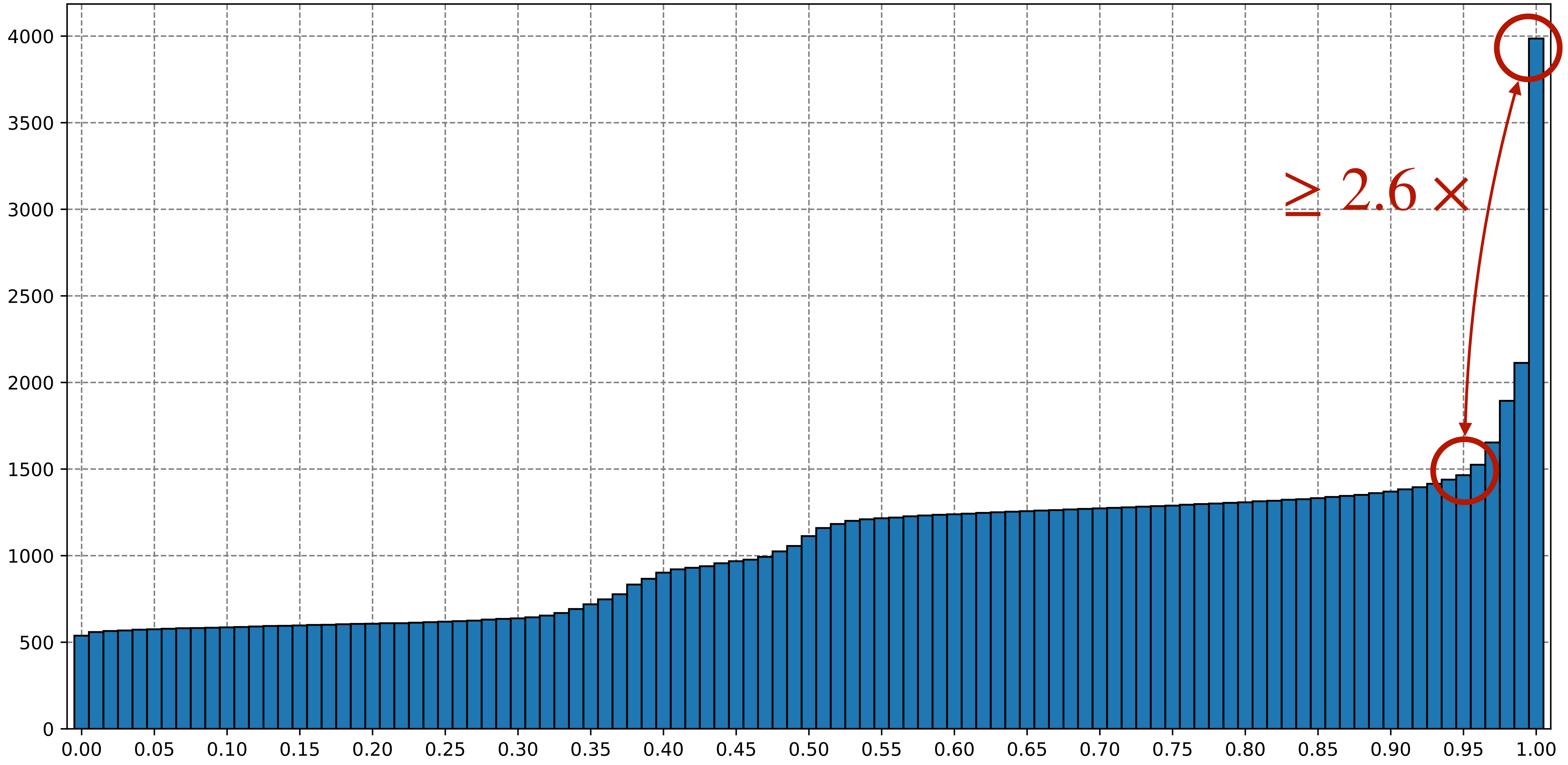


WCET Setting: $\tau_i = (C_i = 4000,\ T_i = 5000,\ D_i = 5000)$

$\Rightarrow$ average processor load: 40%!

B. Brandenburg and M. Gül, "Global scheduling not required: Simple, near-optimal multiprocessor real-time scheduling with semi-partitioned reservations", RTSS 2016

THE NEED FOR BELOW-WORST-CASE PROVISIONING

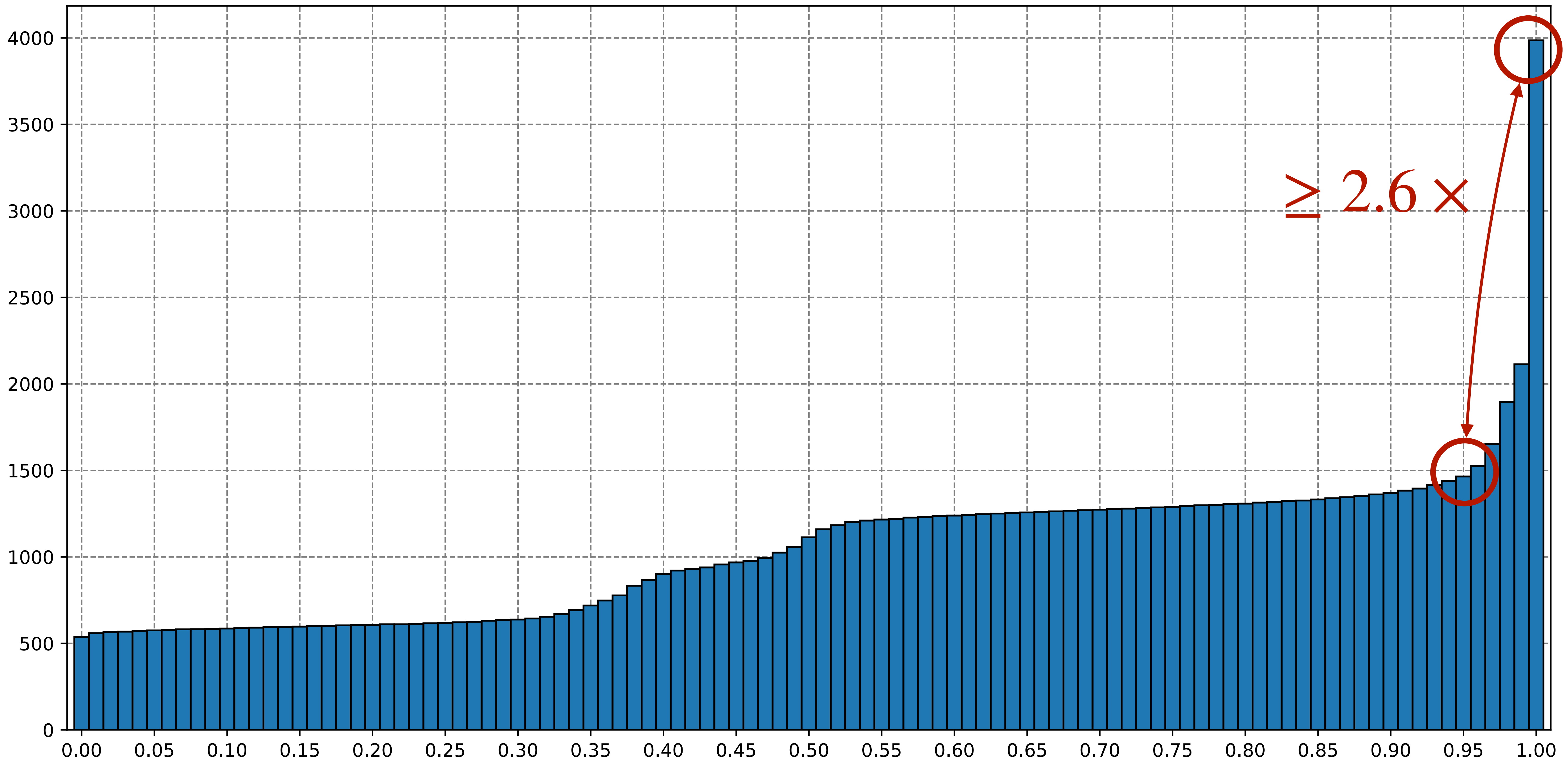


WCET Setting: $\tau_i = (C_i = 4000,\ T_i = 5000,\ D_i = 5000)$ ✖

$\Rightarrow$ average processor load: 40%!

B. Brandenburg and M. Gül, "Global scheduling not required: Simple, near-optimal multiprocessor real-time scheduling with semi-partitioned reservations", RTSS 2016

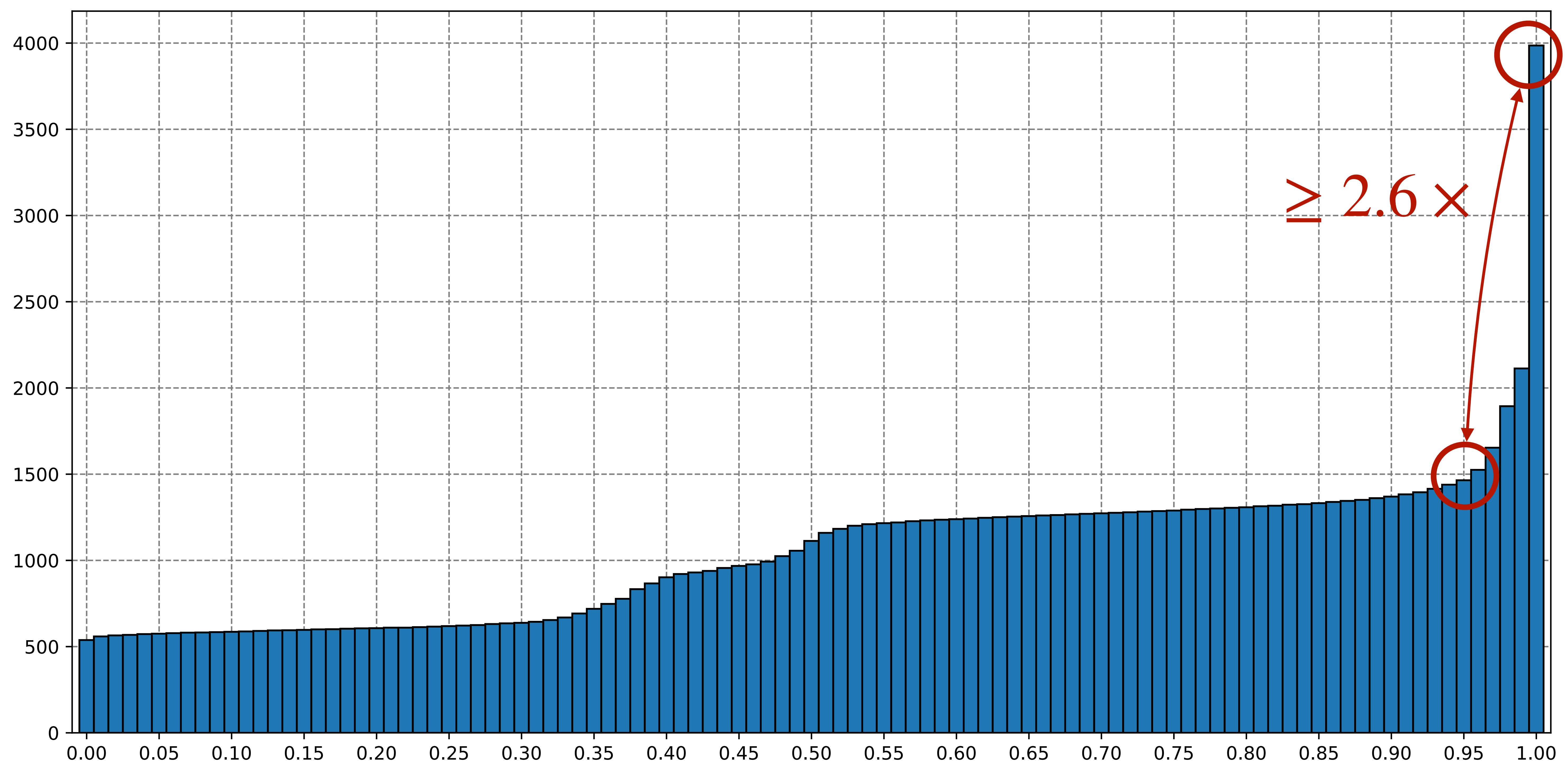
THE NEED FOR BELOW-WORST-CASE PROVISIONING



WCET Setting: $\tau_i = (C_i = 4000,\ T_i = 5000,\ D_i = 5000)$ ✖

B. Brandenburg and M. Gül, "Global scheduling not required: Simple, near-optimal multiprocessor real-time scheduling with semi-partitioned reservations", RTSS 2016

THE NEED FOR BELOW-WORST-CASE PROVISIONING

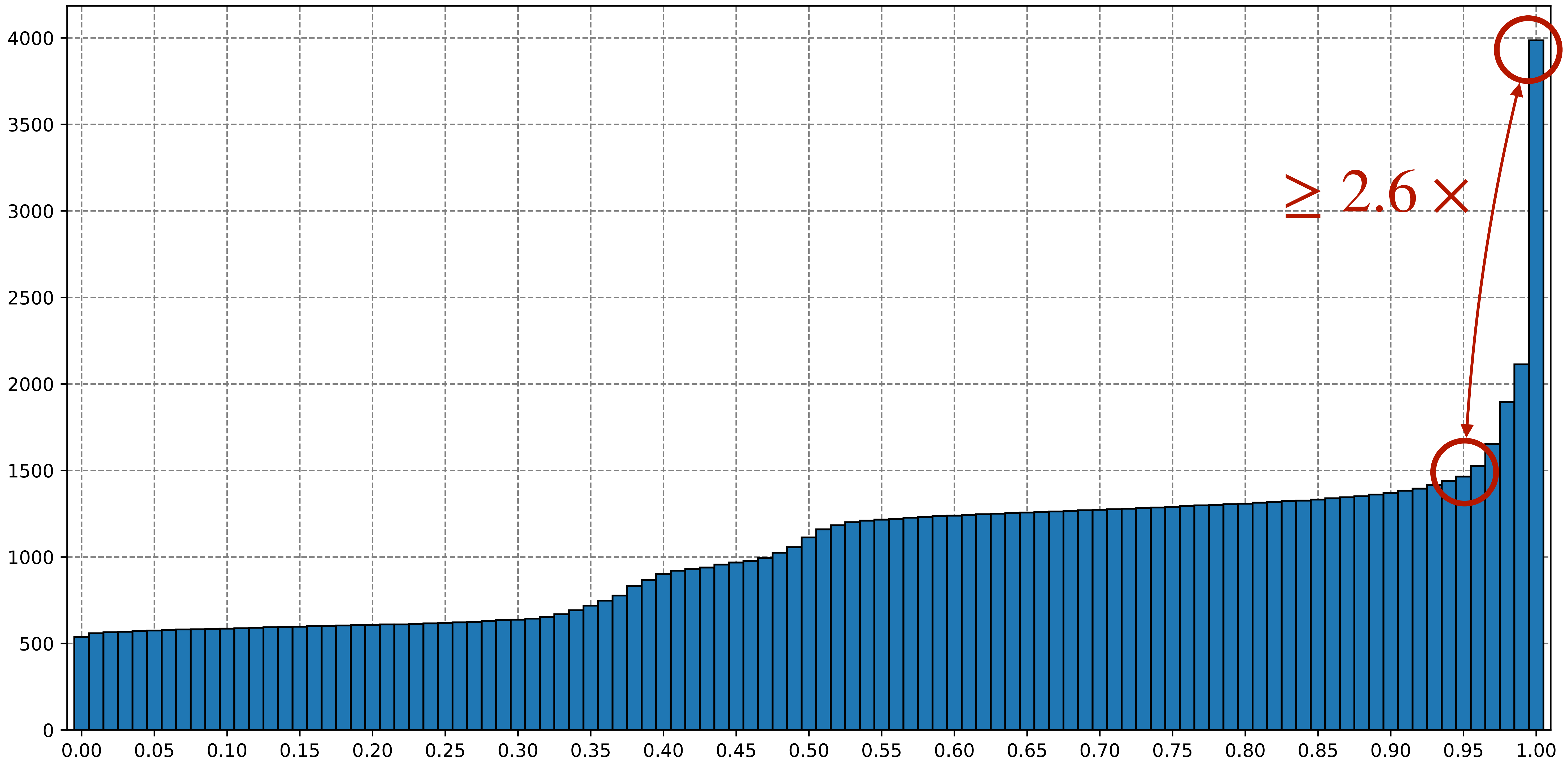


WCET Setting: $\tau_i = (C_i = 4000,\ T_i = 5000,\ D_i = 5000)$ ✖

Prob. Settings: $\tau_i = \left(\mathscr{C}_i = \begin{pmatrix} 1500 & 4000 \\ 0.95 & 0.05 \end{pmatrix},\ T_i = 5000,\ D_i = 5000\right)$

B. Brandenburg and M. Gül, "Global scheduling not required: Simple, near-optimal multiprocessor real-time scheduling with semi-partitioned reservations", RTSS 2016

THE NEED FOR BELOW-WORST-CASE PROVISIONING



WCET Setting: $\tau_i = (C_i = 4000,\ T_i = 5000,\ D_i = 5000)$ ✗

Prob. Settings: $\tau_i = \left(\mathscr{C}_i = \begin{pmatrix} 1500 & 4000 \\ 0.95 & 0.05 \end{pmatrix},\ T_i = 5000,\ D_i = 5000\right)$ ✓

B. Brandenburg and M. Gül, “Global scheduling not required: Simple, near-optimal multiprocessor real-time scheduling with semi-partitioned reservations”, RTSS 2016

PROBABILISTIC RTA

WORST-CASE DEADLINE FAILURE PROBABILITY (WCDFP)

Intuitively: Probability to see the first deadline miss

WORST-CASE DEADLINE FAILURE PROBABILITY (WCDFP)

Intuitively: Probability to see the first deadline miss

$$\Lambda_i := \max_{\xi} \max_{J_{i,j} \in \tau_i} \mathbb{P}\left[\mathscr{R}^{\xi}_{i,j} > D_i\right]$$

WORST-CASE DEADLINE FAILURE PROBABILITY (WCDFP)

Intuitively: Probability to see the first deadline miss

$$\Lambda_i := \max_{\xi} \max_{J_{i,j} \in \tau_i} \mathbb{P}\left[\mathscr{R}^{\xi}_{i,j} > D_i\right]$$

- WCDFP of task τ_i
- Arrival sequence
- j-th activation of task τ_i
- Random variable that describes response time of job $J_{i,j}$ in arrival sequence ξ
- Deadline of task τ_i

WORST-CASE DEADLINE FAILURE PROBABILITY (WCDFP)

Intuitively: Probability to see the first deadline miss

$$\Lambda_i := \max_{\xi} \max_{J_{i,j} \in \tau_i} \mathbb{P}\left[\mathscr{R}^{\xi}_{i,j} > D_i\right]$$

- Λ_i: WCDFP of task τ_i
- ξ: Arrival sequence
- $J_{i,j}$: j-th activation of task τ_i
- $\mathscr{R}^{\xi}_{i,j}$: Random variable that describes response time of job $J_{i,j}$ in arrival sequence ξ
- D_i: Deadline of task τ_i

Pros:

➔ Bounds the expected time to failure of a system

➔ Needed to compute deadline-miss ratio

➔ Worst-case scenario for constrained-deadline tasks under static-priority scheduling: first job under critical-instant pattern

WORST-CASE DEADLINE FAILURE PROBABILITY (WCDFP)

Intuitively: Probability to see the first deadline miss

$$\Lambda_i := \max_{\xi} \max_{J_{i,j} \in \tau_i} \mathbb{P}\left[\mathscr{R}^{\xi}_{i,j} > D_i\right]$$

- WCDFP of task τ_i
- Arrival sequence
- j-th activation of task τ_i
- Random variable that describes response time of job $J_{i,j}$ in arrival sequence ξ
- Deadline of task τ_i

Cons:

➔ Computationally expensive

$$\mathscr{R}^{\xi}_{i,j} := \mathscr{C}_{i,1} + \mathscr{C}_{i,2} + \ldots$$

Probabilistic execution time (pWCET)

n points in distribution

m points in distribution

$\approx n \cdot m$ points in distribution of $\mathscr{C}_{i,1} + \mathscr{C}_{i,2}$

PRIOR WORK ON WCDFP

PRIOR WORK ON WCDFP

Convolution-based approaches:

- ➔ Direct convolution
- ➔ Convolution with re-sampling
- ➔ Task-level convolution

PRIOR WORK ON WCDFP

Convolution-based approaches:

- ➔ Direct convolution
- ➔ Convolution with re-sampling
- ➔ Task-level convolution

Analytical upper-bounds:

- ➔ Bernstein's, Hoeffding's, and Chernoff's inequalities

PRIOR WORK ON WCDFP

Convolution-based approaches:

- ➔ Direct convolution
- ➔ Convolution with re-sampling
- ➔ Task-level convolution

Analytical upper-bounds:

- ➔ Bernstein's, Hoeffding's, and Chernoff's inequalities

Common disadvantages:

PRIOR WORK ON WCDFP

Convolution-based approaches:

- ➔ Direct convolution
- ➔ Convolution with re-sampling
- ➔ Task-level convolution

Analytical upper-bounds:

- ➔ Bernstein's, Hoeffding's, and Chernoff's inequalities

Common disadvantages:

- ➔ Highly-dependent on the input

PRIOR WORK ON WCDFP

Convolution-based approaches:

- ➔ Direct convolution
- ➔ Convolution with re-sampling
- ➔ Task-level convolution

Analytical upper-bounds:

- ➔ Bernstein's, Hoeffding's, and Chernoff's inequalities

Common disadvantages:

- ➔ Highly-dependent on the input
- ➔ Methods to bound pessimism are unknown

PRIOR WORK ON WCDFP

Convolution-based approaches:

- ➔ Direct convolution
- ➔ Convolution with re-sampling
- ➔ Task-level convolution

Analytical upper-bounds:

- ➔ Bernstein's, Hoeffding's, and Chernoff's inequalities

Common disadvantages:

- ➔ Highly-dependent on the input
- ➔ Methods to bound pessimism are unknown
- ➔ Hard to guess the right parameters

MONTE CARLO
WCDFP ESTIMATION

MAIN STEPS OF OUR APPROACH

1. Change the problem statement

⇒

2. Sample many values from response-time distribution $\mathscr{R}^{\xi}_{i,j}$

⇒

3. Perform statistical generalization to estimate WCDFP Λ_i

Standard idea of Monte Carlo

MAIN STEPS OF OUR APPROACH

1. Change the problem statement

$\Rightarrow$

2. Sample many values from response-time distribution $\mathscr{R}_{i,j}^{\xi}$

$\Rightarrow$

3. Perform statistical generalization to estimate WCDFP Λ_i

MAIN STEPS OF OUR APPROACH

1. Change the problem statement

$\Rightarrow$

2. Sample many values from response-time distribution $\mathscr{R}_{i,j}^{\xi}$

$\Rightarrow$

3. Perform statistical generalization to estimate WCDFP Λ_i

1. CHANGING THE PROBLEM STATEMENT

$\mathscr{R}^{\xi}_{i,j}$ – response time

D_i – deadline of τ_i

Prior statement:

Given a task set τ, a task τ_i, arrival sequence ξ, and a job $J_{i,j}$, derive an upper bound r such that $\mathbb{P}[\mathscr{R}^{\xi}_{i,j} > D_i] \leq r$

1. CHANGING THE PROBLEM STATEMENT

$\mathscr{R}^{\xi}_{i,j}$ – response time

D_i – deadline of τ_i

δ – accuracy

ε – misestimation probability

Prior statement:

Given a task set τ, a task τ_i, arrival sequence ξ, and a job $J_{i,j}$, derive an upper bound r such that $\mathbb{P}[\mathscr{R}^{\xi}_{i,j} > D_i] \leq r$

New statement:

Given a task set τ, a task τ_i, arrival sequence ξ, a job $J_{i,j}$,
the required accuracy δ, and the misestimation probability ε,
derive ~~an upper~~ bounds l and r such that
$l \leq \mathbb{P}[\mathscr{R}^{\xi}_{i,j} > D_i] \leq r$ with probability $1 - \varepsilon$ and $|r - l| < \delta$

MAIN STEPS OF OUR APPROACH

1. Change the problem statement

⇒

2. Sample many values from response-time distribution $\mathcal{R}^{\xi}_{i,j}$

⇒

3. Perform statistical generalization to estimate WCDFP Λ_i

2. SAMPLE MANY VALUES FROM $\mathscr{R}^{\xi}_{i,j}$

$\mathscr{R}^{\xi}_{i,j}$ – response time

$\mathsf{S}^{\xi}_{i,j}$ – simulator

Recall: distribution of $\mathscr{R}^{\xi}_{i,j}$ (likely) contains too many points

$\implies$ we cannot compute the distribution

2. SAMPLE MANY VALUES FROM $\mathscr{R}^{\xi}_{i,j}$

$\mathscr{R}^{\xi}_{i,j}$ – response time

$\mathsf{S}^{\xi}_{i,j}$ – simulator

Recall: distribution of $\mathscr{R}^{\xi}_{i,j}$ (likely) contains too many points

$\Longrightarrow$ we cannot compute the distribution

However! We still can build a procedure to sample values from $\mathscr{R}^{\xi}_{i,j}$

In the paper: A simple schedule simulator $\mathsf{S}^{\xi}_{i,j}$ does the job

2. SAMPLE MANY VALUES FROM $\mathscr{R}^{\xi}_{i,j}$

$\mathscr{R}^{\xi}_{i,j}$ – response time

$\mathsf{S}^{\xi}_{i,j}$ – simulator

Recall: distribution of $\mathscr{R}^{\xi}_{i,j}$ (likely) contains too many points

$\Longrightarrow$ we cannot compute the distribution

However! We still can build a procedure to sample values from $\mathscr{R}^{\xi}_{i,j}$

In the paper: A simple schedule simulator $\mathsf{S}^{\xi}_{i,j}$ does the job

Theorem: distribution of $\mathsf{S}^{\xi}_{i,j}$ = distribution of $\mathscr{R}^{\xi}_{i,j}$

MAIN STEPS OF OUR APPROACH

1. Change the problem statement

⇒

2. Sample many values from response-time distribution $\mathscr{R}_{i,j}^{\xi}$

⇒

3. Perform statistical generalization to estimate WCDFP Λ_i

3. PERFORM STATISTICAL GENERALIZATION

ξ – arrival sequence

δ – accuracy

ε – misestimation probability

$\mathscr{R}^{\xi}_{x,y}$ – response time

D_x – deadline of τ_x

Algorithm 1: DFP estimation

Input: τ, τ_x, ξ, δ, and ε.

Output: Estimate of $\mathbb{P}\left[\mathcal{R}^{\xi}_{x,y} > D_x\right]$.

3. PERFORM STATISTICAL GENERALIZATION

ξ – arrival sequence

δ – accuracy

ε – misestimation probability

$\mathcal{R}^{\xi}_{x,y}$ – response time

D_x – deadline of τ_x

Algorithm 1: DFP estimation

Input: τ, τ_x, ξ, δ, and ε.

Output: Estimate of $\mathbb{P}\left[\mathcal{R}^{\xi}_{x,y} > D_x\right]$.

Input Parameters

Note: δ and ε are explicit arguments!

3. PERFORM STATISTICAL GENERALIZATION

ξ – arrival sequence

δ – accuracy

ε – misestimation probability

$\mathcal{R}^{\xi}_{x,y}$ – response time

D_x – deadline of τ_x

Algorithm 1: DFP estimation

Input: τ, τ_x, ξ, δ, and ε.

Output: Estimate of $\mathbb{P}\left[\mathcal{R}^{\xi}_{x,y} > D_x\right]$.

Input Parameters

Note: δ and ε are explicit arguments!

$\Longrightarrow$ Easy to chose right parameters

3. PERFORM STATISTICAL GENERALIZATION

ξ – arrival sequence

δ – accuracy

ε – misestimation probability

$\mathscr{R}^{\xi}_{x,y}$ – response time

D_x – deadline of τ_x

Algorithm 1: DFP estimation

Input: τ, τ_x, ξ, δ, and ε.

Output: Estimate of $\mathbb{P}\left[\mathcal{R}^{\xi}_{x,y} > D_x\right]$.

1 $k := 0$, $z := \Phi^{-1}\left(1 - \frac{\varepsilon}{2}\right)$, $s := \lceil (z/\delta)^2 \rceil$;

Input Parameters

Note: δ and ε are explicit arguments!

Number of necessary samples s depends only on δ and ε

3. PERFORM STATISTICAL GENERALIZATION

ξ – arrival sequence

δ – accuracy

ε – misestimation probability

$\mathscr{R}^{\xi}_{x,y}$ – response time

D_x – deadline of τ_x

Algorithm 1: DFP estimation

Input: τ, τ_x, ξ, δ, and ε.

Output: Estimate of $\mathbb{P}\left[\mathcal{R}^{\xi}_{x,y} > D_x\right]$.

1 $k := 0$, $z := \Phi^{-1}\left(1 - \frac{\varepsilon}{2}\right)$, $s := \lceil (z/\delta)^2 \rceil$;

Input Parameters

Note: δ and ε are explicit arguments!

Number of necessary samples s depends only on δ and ε

z is $(1 - \varepsilon/2)$-th quantile of standard normal distribution

s is any number greater than $(z/\delta)^2$

3. PERFORM STATISTICAL GENERALIZATION

ξ – arrival sequence

δ – accuracy

ε – misestimation probability

$\mathscr{R}^{\xi}_{x,y}$ – response time

D_x – deadline of τ_x

Algorithm 1: DFP estimation

Input: τ, τ_x, ξ, δ, and ε.

Output: Estimate of $\mathbb{P}\left[\mathcal{R}^{\xi}_{x,y} > D_x\right]$.

1 $k := 0$, $z := \Phi^{-1}\left(1 - \frac{\varepsilon}{2}\right)$, $s := \lceil (z/\delta)^2 \rceil$;

Input Parameters

Note: δ and ε are explicit arguments!

Number of necessary samples s depends only on δ and ε

z is $(1 - \varepsilon/2)$-th quantile of standard normal distribution

s is any number greater than $(z/\delta)^2$

$\implies$ Runtime depends on δ, ε, and runtime of simulator

3. PERFORM STATISTICAL GENERALIZATION

ξ – arrival sequence

δ – accuracy

ε – misestimation probability

$\mathcal{R}^{\xi}_{x,y}$ – response time

D_x – deadline of τ_x

$\mathsf{S}^{\xi}_{x,y}$ – simulator

Algorithm 1: DFP estimation

Input: τ, τ_x, ξ, δ, and ε.

Output: Estimate of $\mathbb{P}\left[\mathcal{R}^{\xi}_{x,y} > D_x\right]$.

1 $k := 0$, $z := \Phi^{-1}\left(1 - \frac{\varepsilon}{2}\right)$, $s := \lceil (z/\delta)^2 \rceil$;

2 **for** 1 **to** s **do**

3   Draw sample via $\mathsf{S}^{\xi}_{x,y}$

4   **if** $\mathsf{S}^{\xi}_{x,y} > D_x$ **then**

5     $k := k + 1$;

Input Parameters

Note: δ and ε are explicit arguments!

Number of necessary samples s depends only on δ and ε

Do s simulations and count the number of deadline misses k

3. PERFORM STATISTICAL GENERALIZATION

- ξ – arrival sequence
- δ – accuracy
- ε – misestimation probability
- $\mathscr{R}^{\xi}_{x,y}$ – response time
- D_x – deadline of τ_x
- $\mathsf{S}^{\xi}_{x,y}$ – simulator

Algorithm 1: DFP estimation

Input: τ, τ_x, ξ, δ, and ε.

Output: Estimate of $\mathbb{P}\left[\mathcal{R}^{\xi}_{x,y} > D_x\right]$.

1 $k := 0$, $z := \Phi^{-1}\left(1 - \frac{\varepsilon}{2}\right)$, $s := \lceil (z/\delta)^2 \rceil$;

2 **for** 1 **to** s **do**

3 Draw sample via $\mathsf{S}^{\xi}_{x,y}$

4 **if** $\mathsf{S}^{\xi}_{x,y} > D_x$ **then**

5 $k := k + 1$;

6 $\tilde{s} := s + z^2$, $\tilde{p} := \frac{1}{\tilde{s}}\left(k + \frac{z^2}{2}\right)$;

7 **return** $\tilde{p} \pm z\sqrt{\frac{\tilde{p}(1-\tilde{p})}{\tilde{s}}}$

Input Parameters

Note: δ and ε are explicit arguments!

Number of necessary samples s depends only on δ and ε

Do s simulations and count the number of deadline misses k

Given k successes in s trials, one can estimate the ground truth p

THERE IS MORE

Introduce formal definition of probabilistic response-time $\mathscr{R}^{\xi}_{i,j}$

➔ Can be used in future work

Correctness of the simulator

➔ Detailed proof that interprets simulator as random variable

Correctness of statistical generalization

➔ Reduction of the simulation to a Bernoulli trial

➔ Application of binomial confidence interval

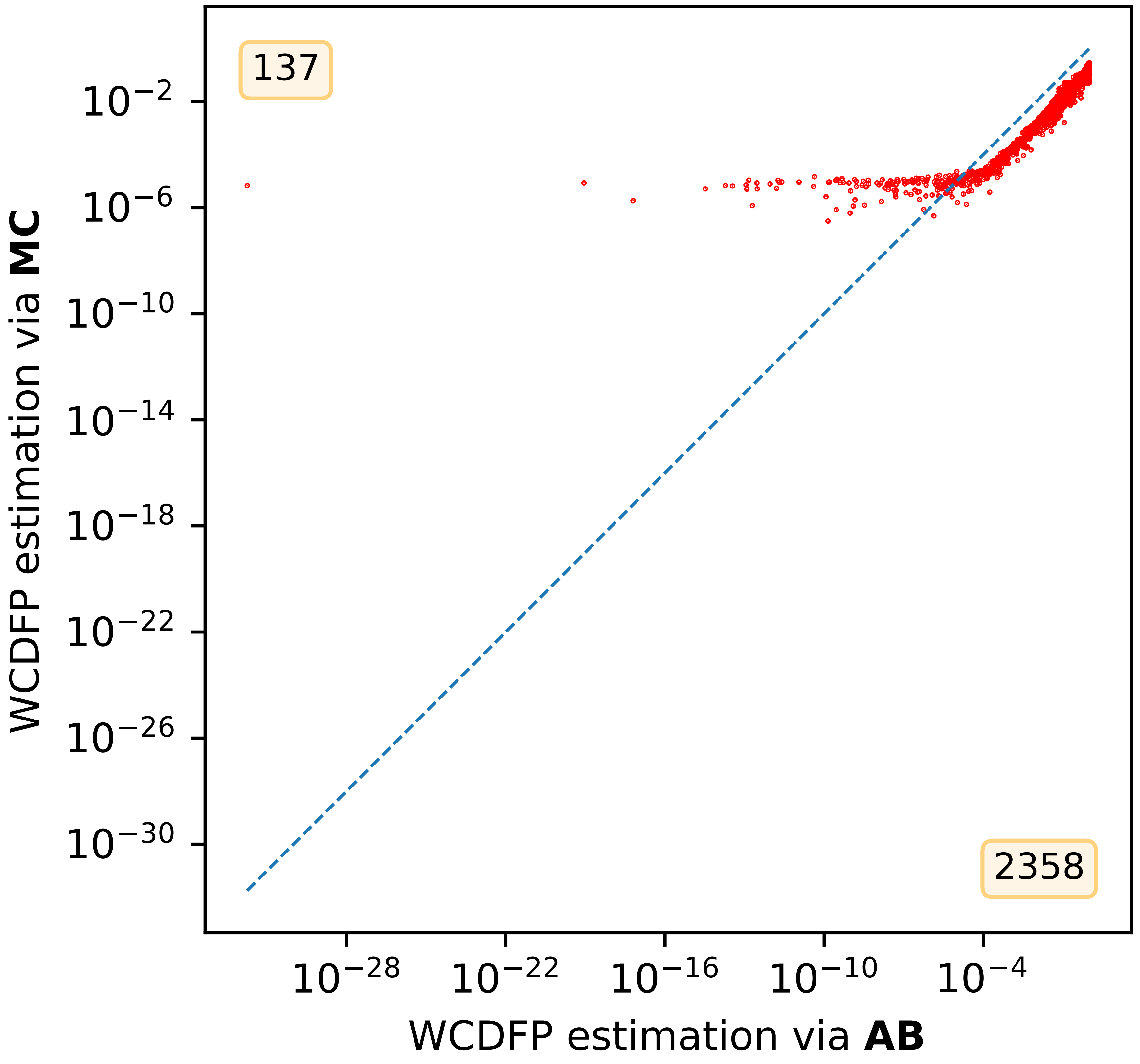
Evaluation:

… will be discussed next

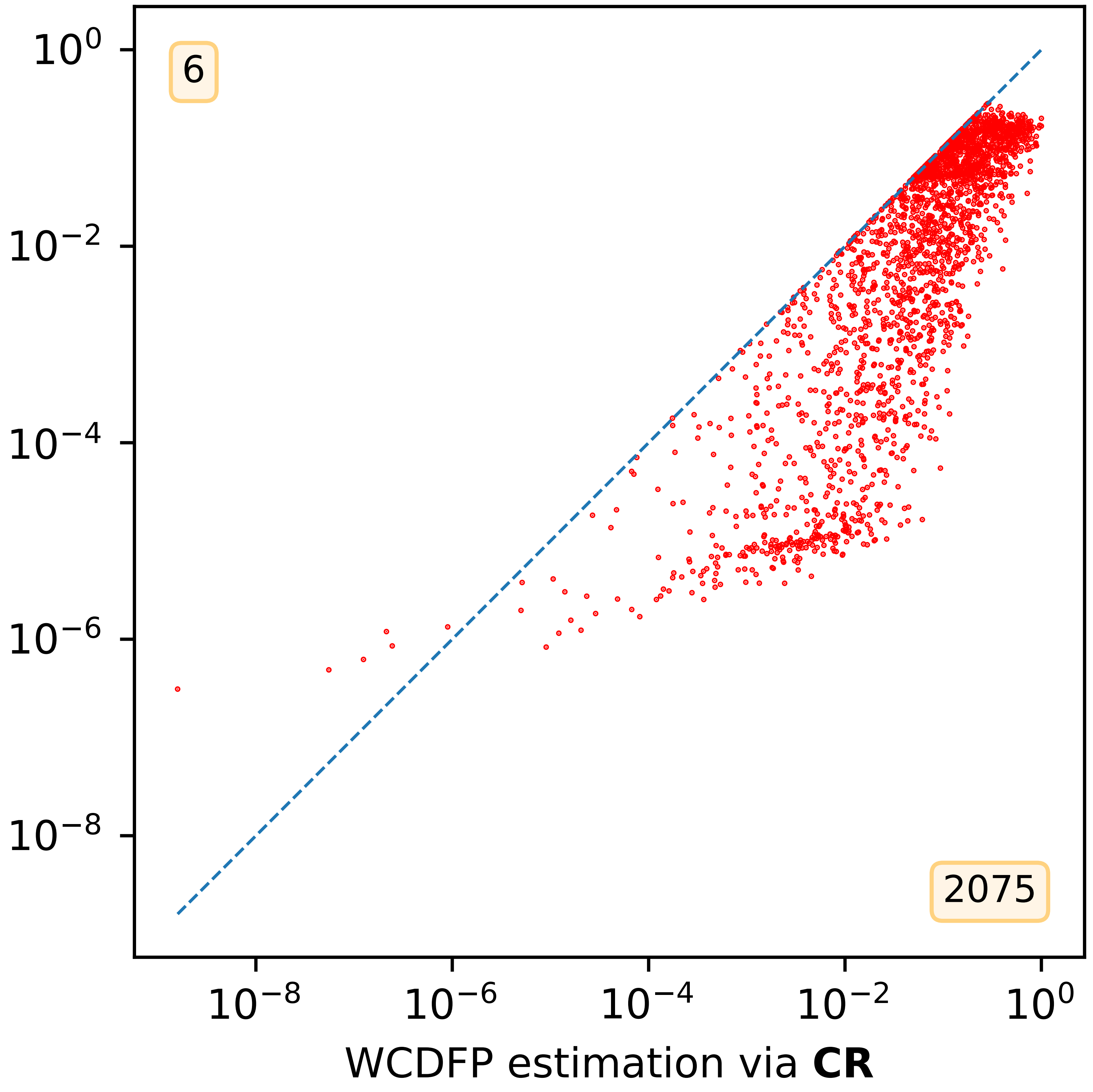
EVALUATION

EVALUATION

Chernoff's inequality vs MC



Convolution with re-sampling vs MC



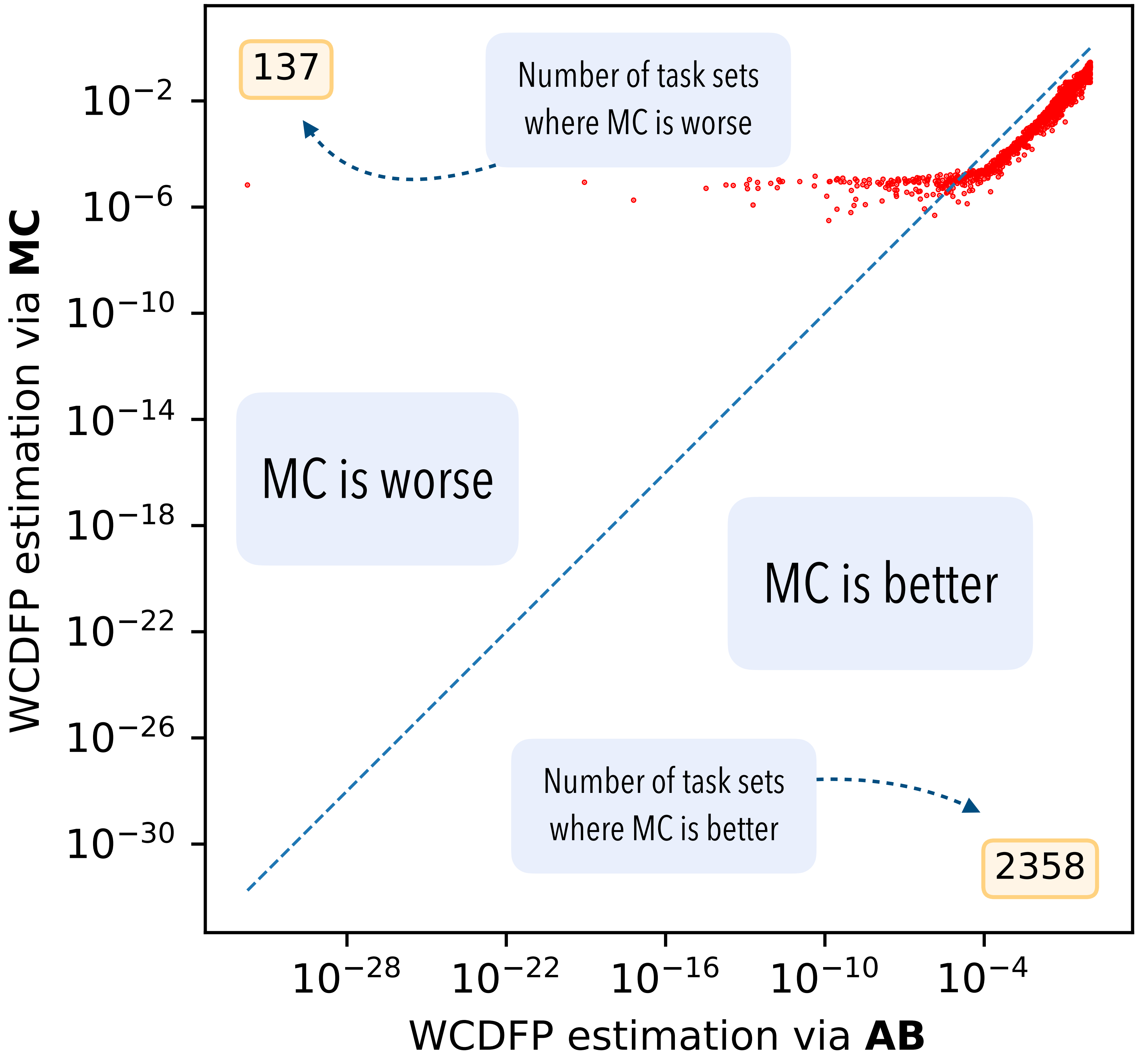
- 2500 task sets
- Shape: $\mathscr{C}_i = \begin{pmatrix} c & 4c \\ 0.95 & 0.05 \end{pmatrix}$

Probabilistic execution time (pWCET)

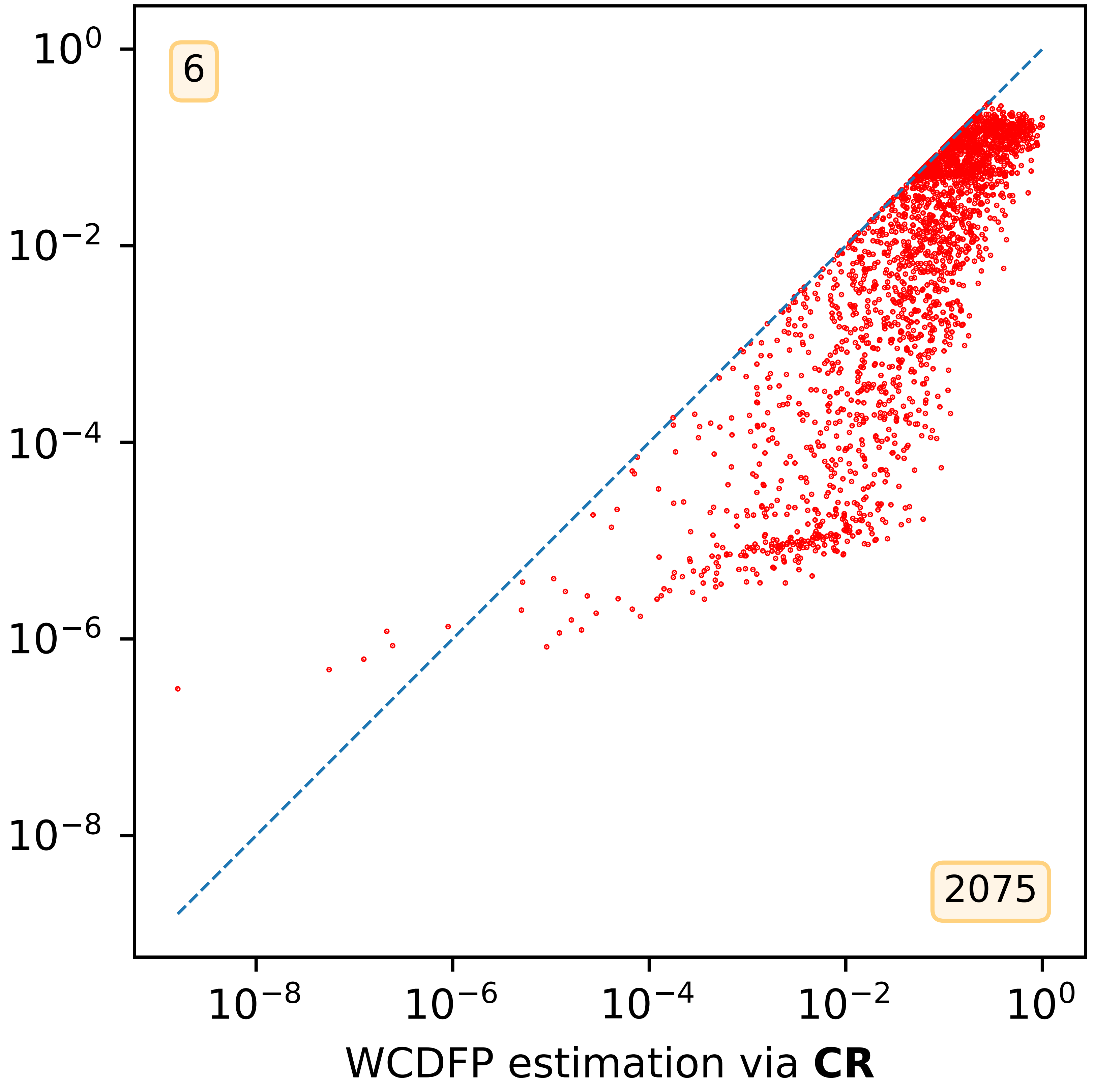
- Cardinality: $n \in \{5, 10, \ldots 50\}$
- Utilization: $u \in \{0.75, 0.8, \ldots, 0.95\}$

EVALUATION

Chernoff's inequality vs MC



Convolution with re-sampling vs MC



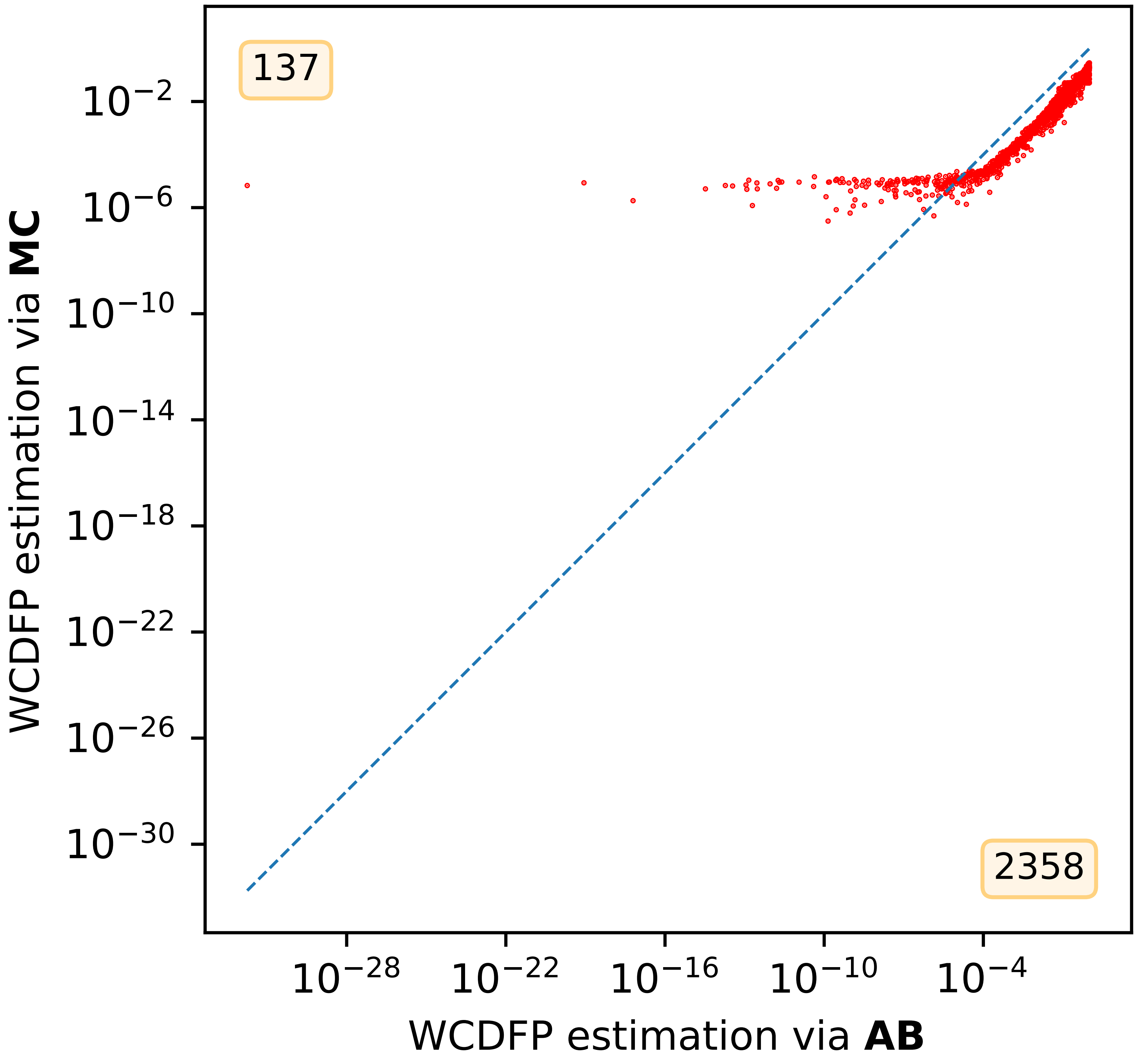
- 2500 task sets
- Shape: $\mathscr{C}_i = \begin{pmatrix} c & 4c \\ 0.95 & 0.05 \end{pmatrix}$

Probabilistic execution time (pWCET)

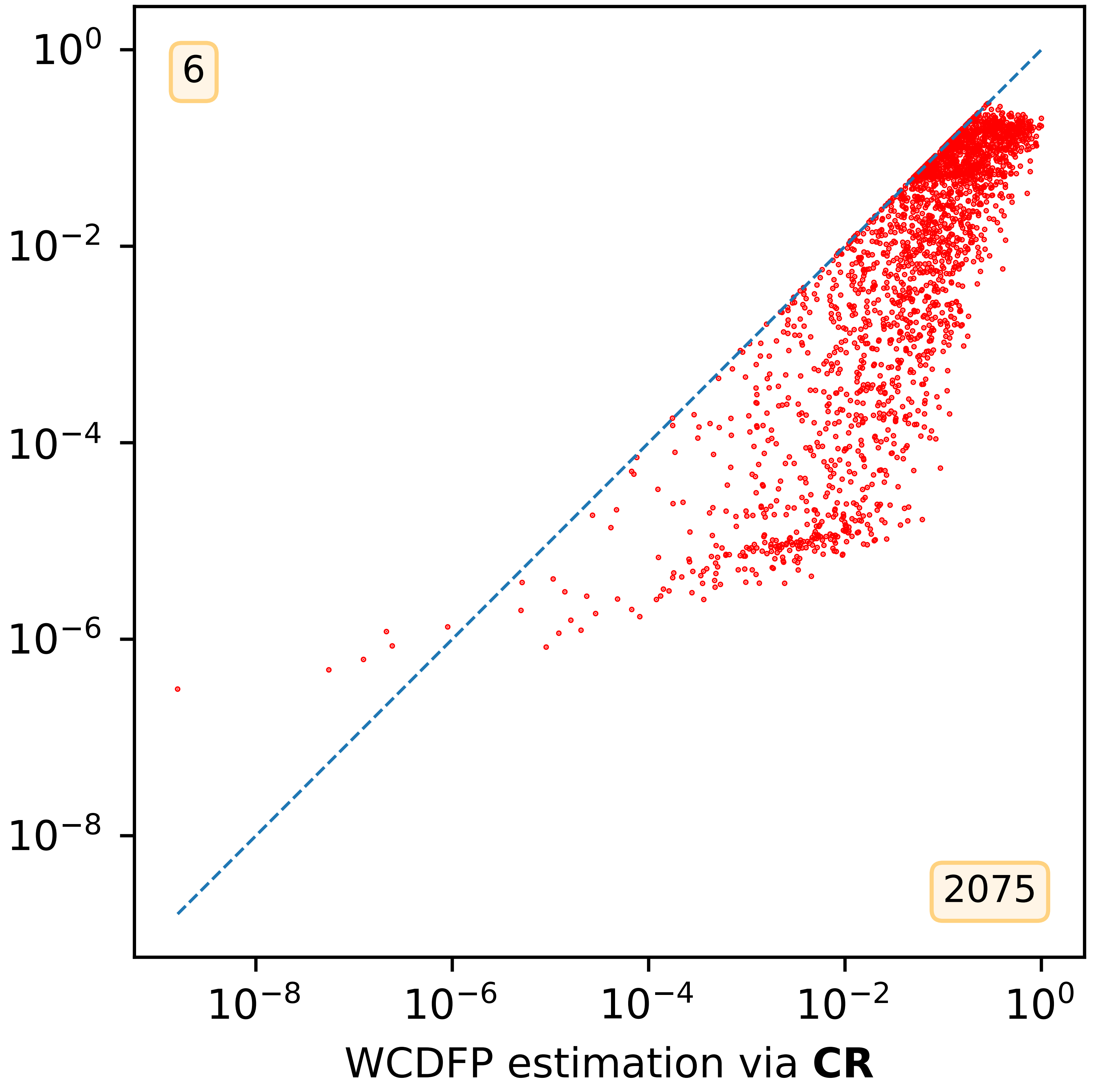
- Cardinality: $n \in \{5, 10, \ldots 50\}$
- Utilization: $u \in \{0.75, 0.8, \ldots, 0.95\}$

EVALUATION

Chernoff's inequality vs MC



Convolution with re-sampling vs MC



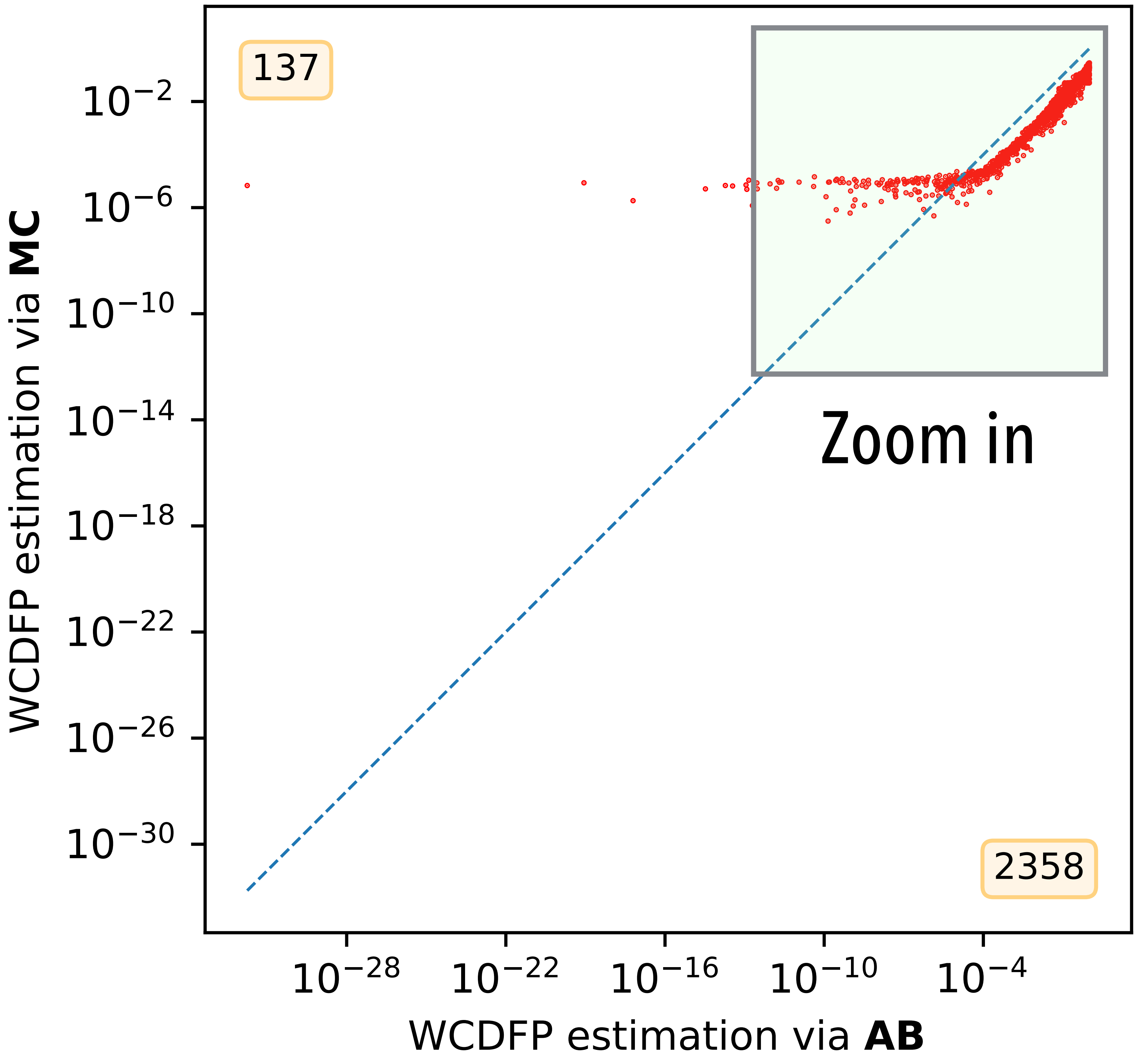
- ➔ 2500 task sets
- ➔ Shape: $\mathscr{C}_i = \begin{pmatrix} c & 4c \\ 0.95 & 0.05 \end{pmatrix}$

Probabilistic execution time (pWCET)

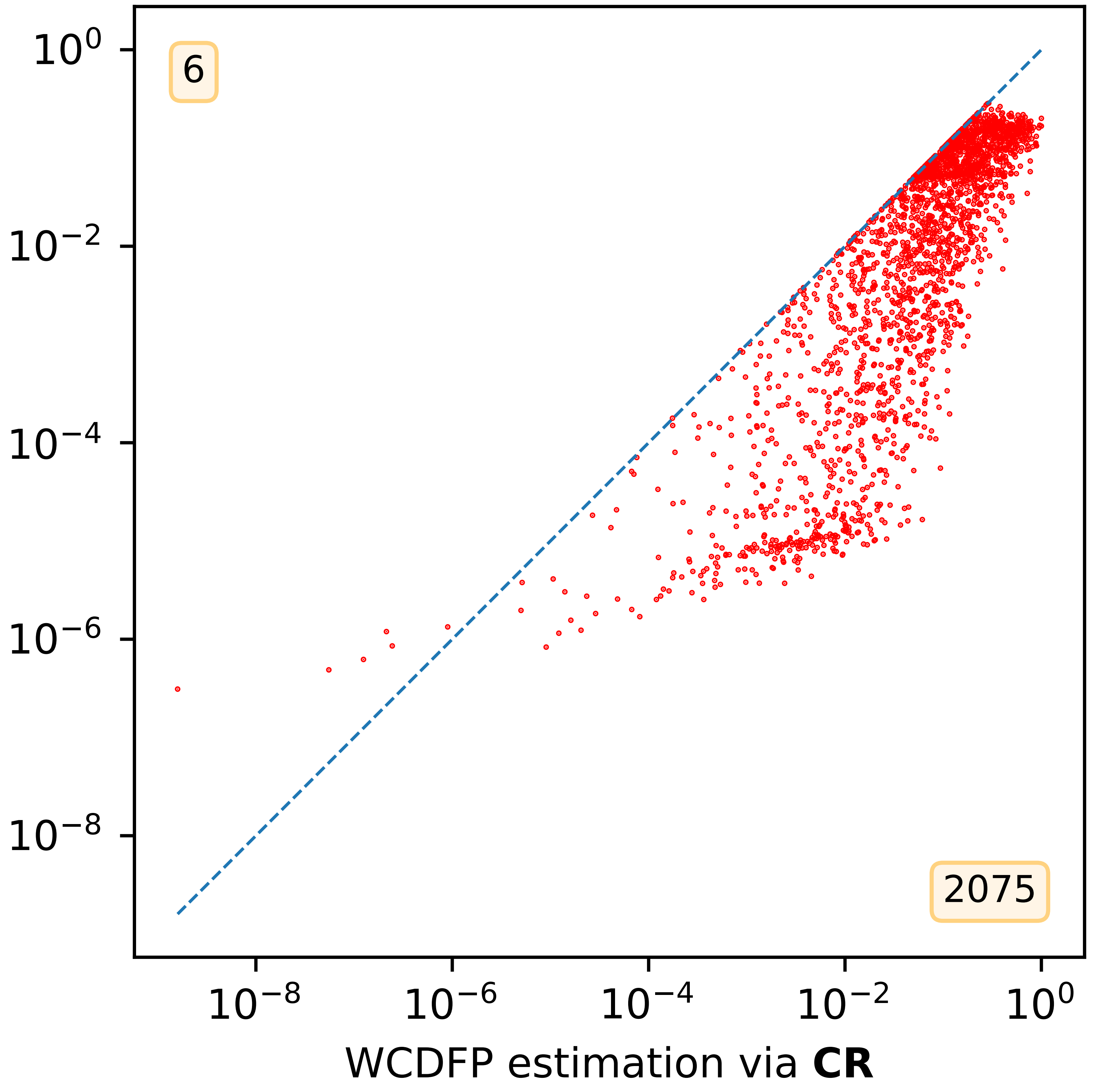
- ➔ Cardinality: $n \in \{5, 10, \ldots 50\}$
- ➔ Utilization: $u \in \{0.75, 0.8, \ldots, 0.95\}$

EVALUATION

Chernoff's inequality vs MC



Convolution with re-sampling vs MC



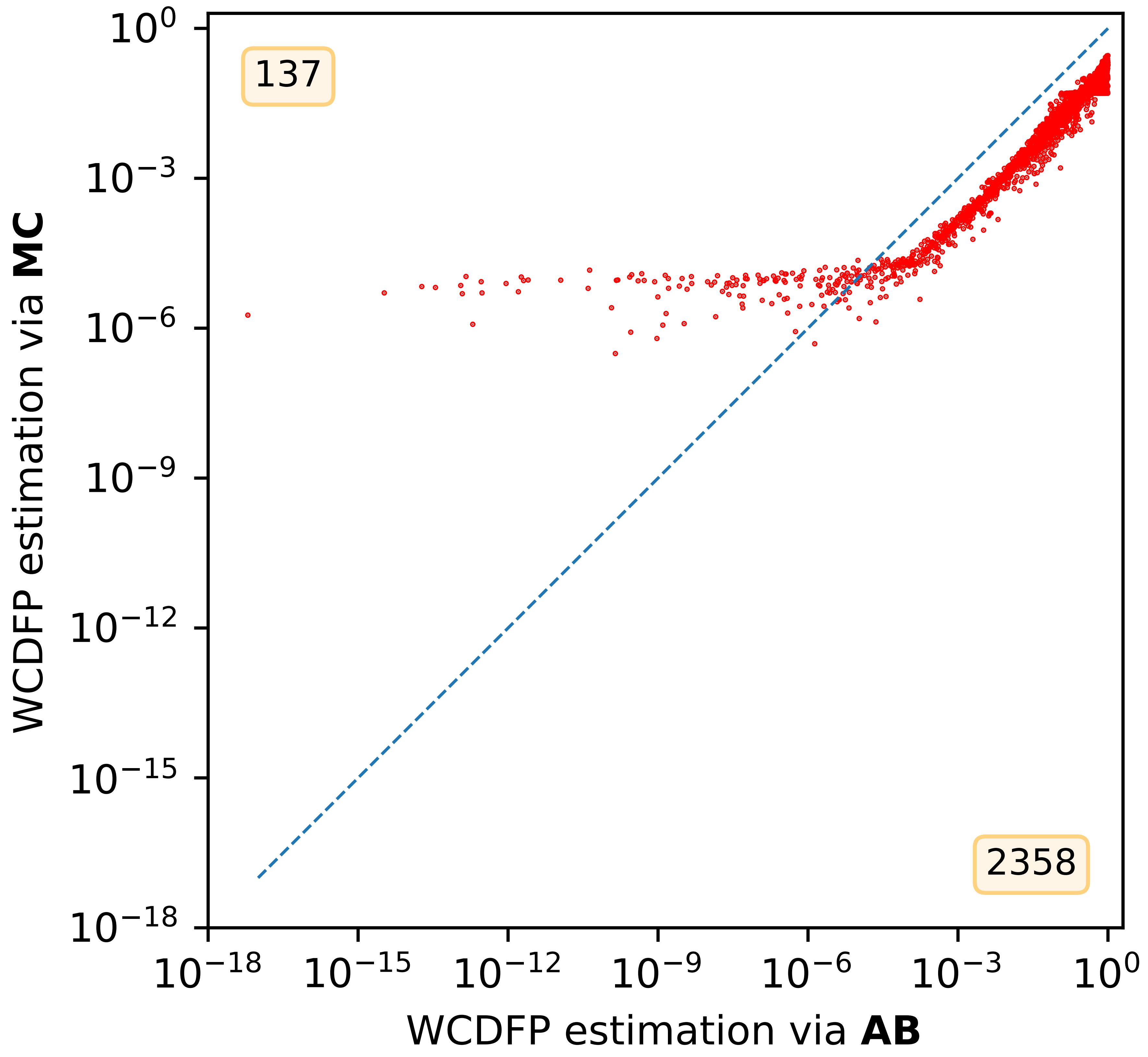
- ➔ 2500 task sets
- ➔ Shape: $\mathscr{C}_i = \begin{pmatrix} c & 4c \\ 0.95 & 0.05 \end{pmatrix}$

Probabilistic execution time (pWCET)

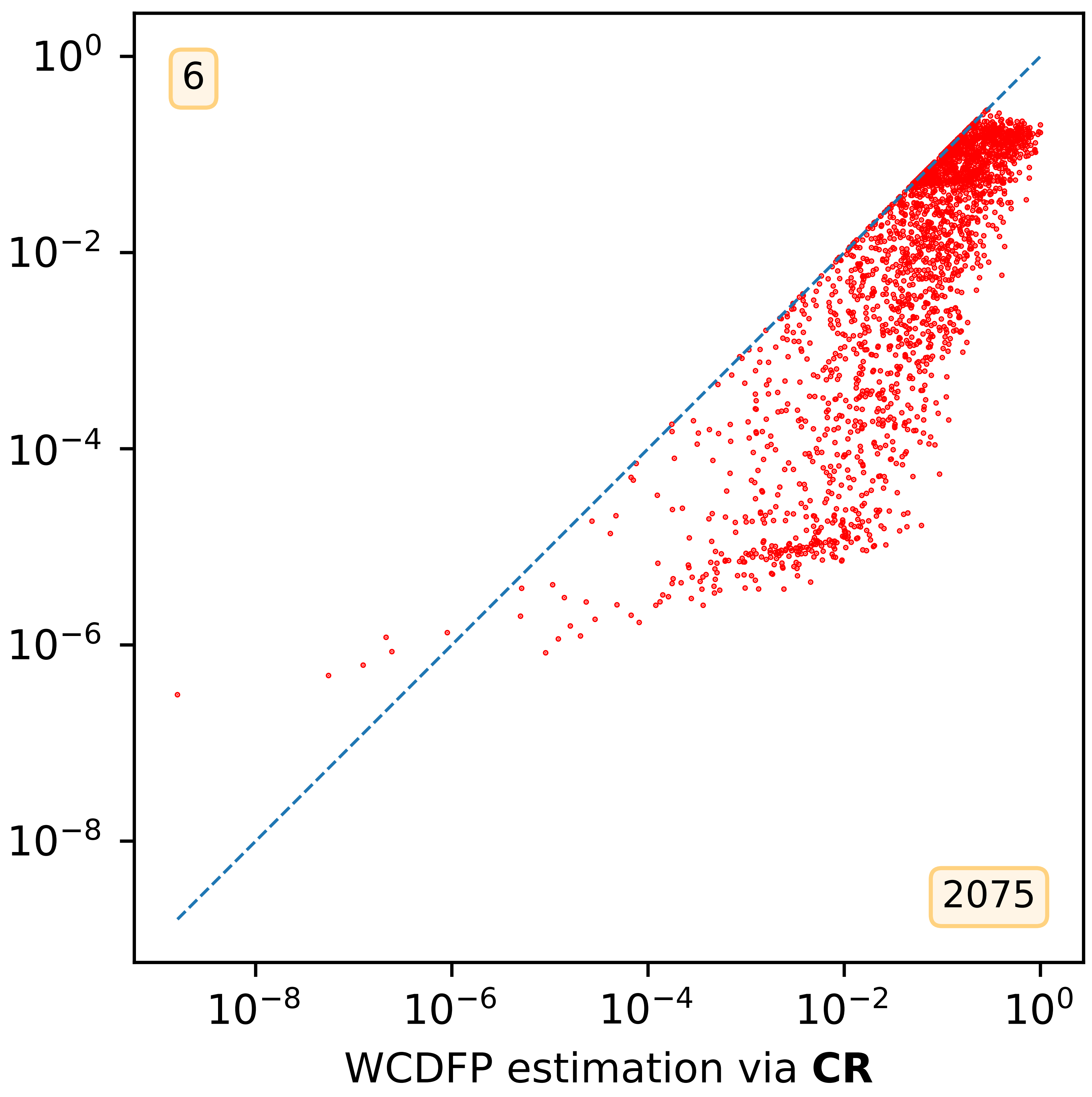
- ➔ Cardinality: $n \in \{5,10,\ldots 50\}$
- ➔ Utilization: $u \in \{0.75,0.8,\ldots,0.95\}$

EVALUATION

Chernoff's inequality vs MC



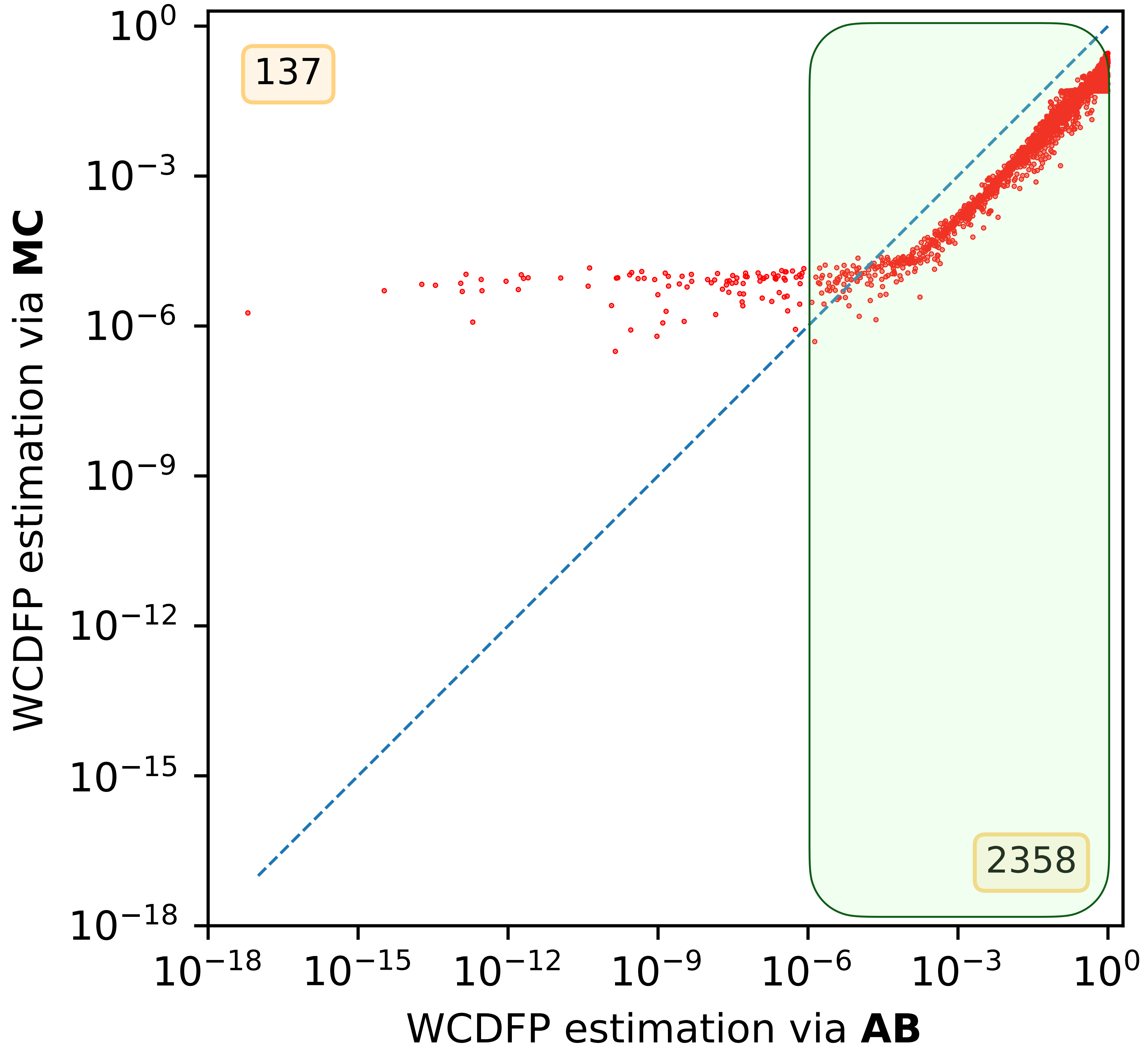
Convolution with re-sampling vs MC



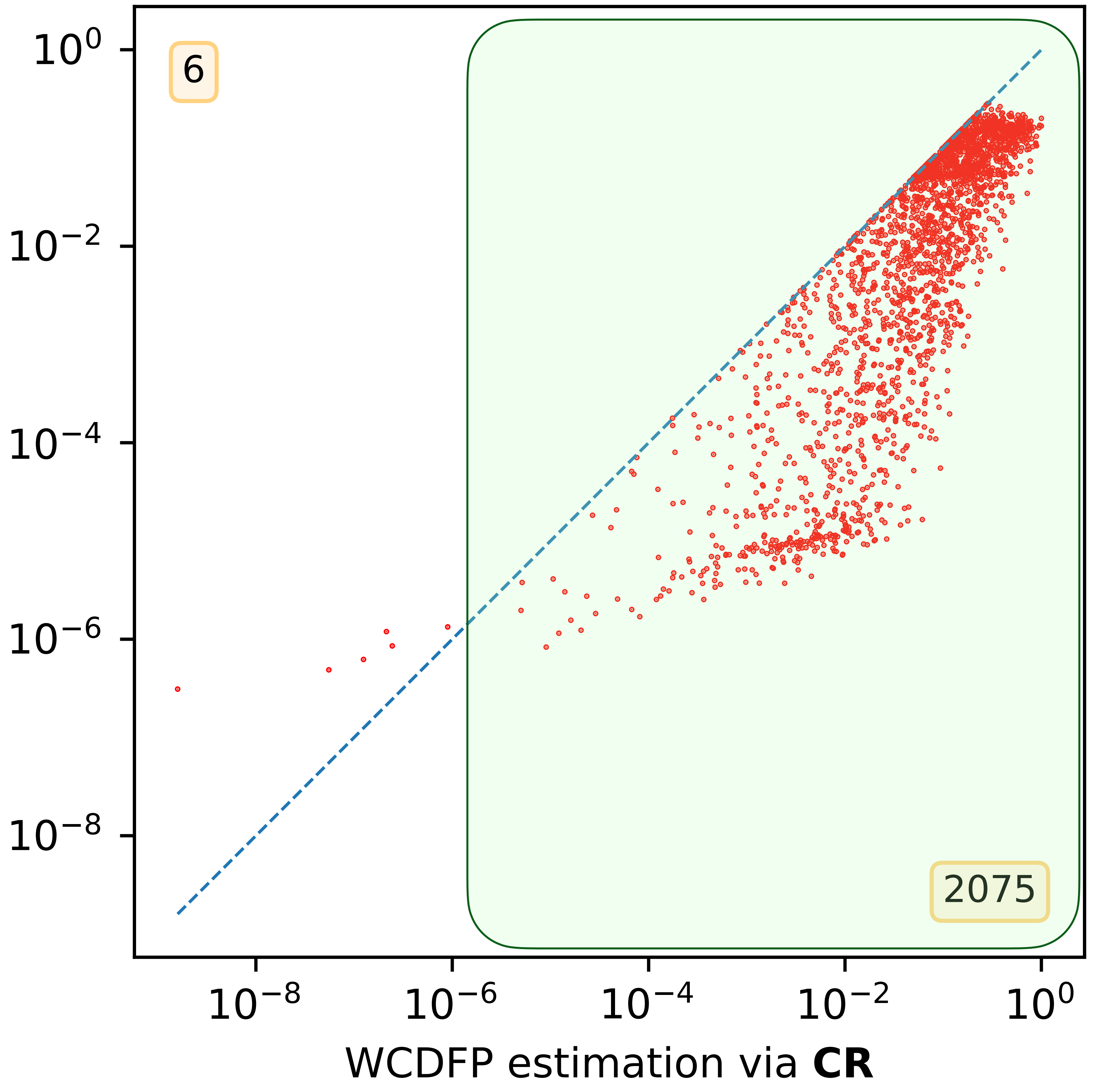
- 2500 task sets
- Shape: $\mathscr{C}_i = \begin{pmatrix} c & 4c \\ 0.95 & 0.05 \end{pmatrix}$ (Probabilistic execution time (pWCET))
- Cardinality: $n \in \{5, 10, \ldots 50\}$
- Utilization: $u \in \{0.75, 0.8, \ldots, 0.95\}$

EVALUATION

Chernoff's inequality vs MC



Convolution with re-sampling vs MC



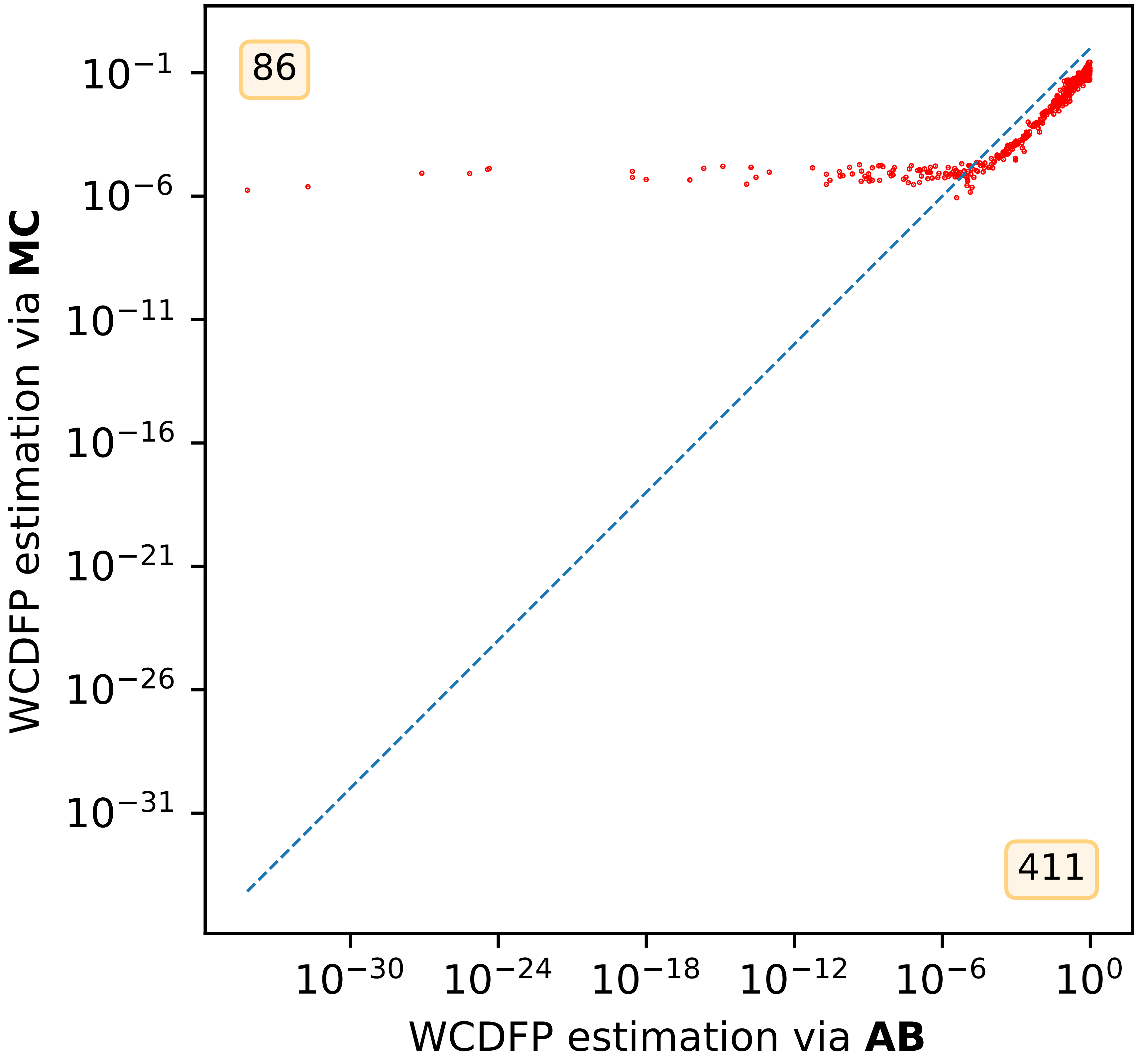
- ➔ 2500 task sets
- ➔ Shape: $\mathscr{C}_i = \begin{pmatrix} c & 4c \\ 0.95 & 0.05 \end{pmatrix}$

Probabilistic execution time (pWCET)

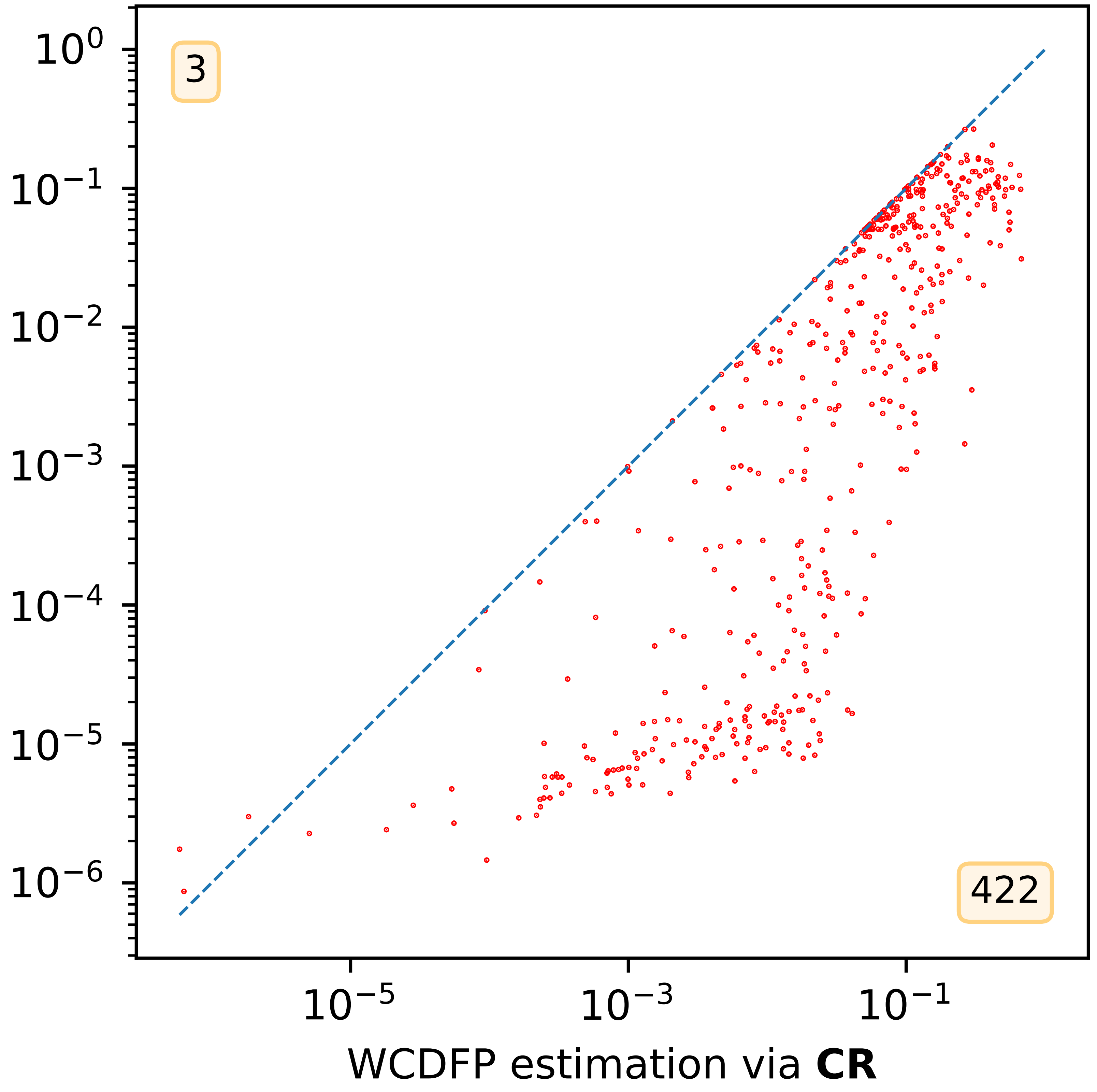
- ➔ Cardinality: $n \in \{5,10,\ldots 50\}$
- ➔ Utilization: $u \in \{0.75,0.8,\ldots,0.95\}$

EVALUATION (VARY COST DIFFERENCE)

Chernoff's inequality vs MC

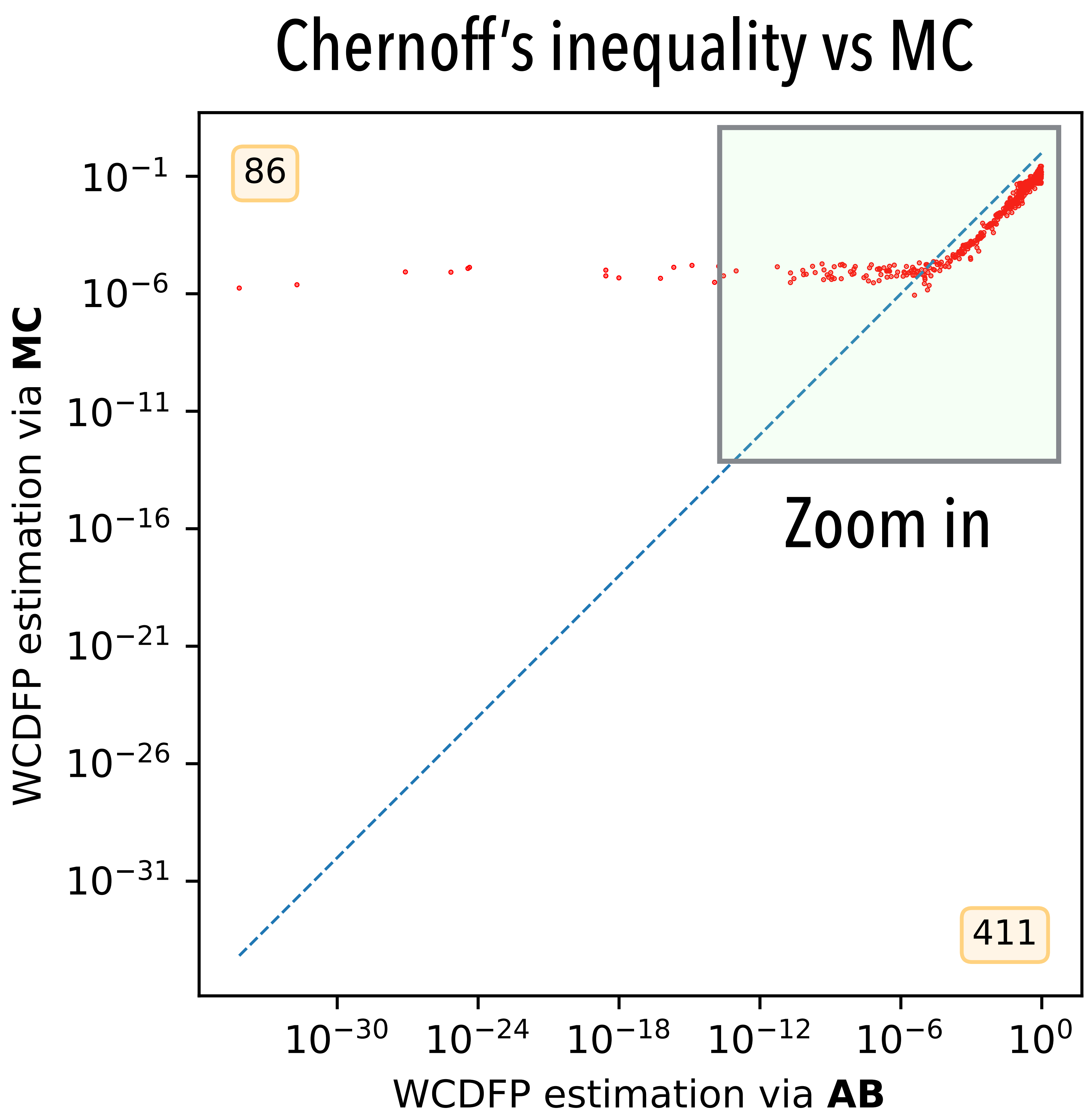


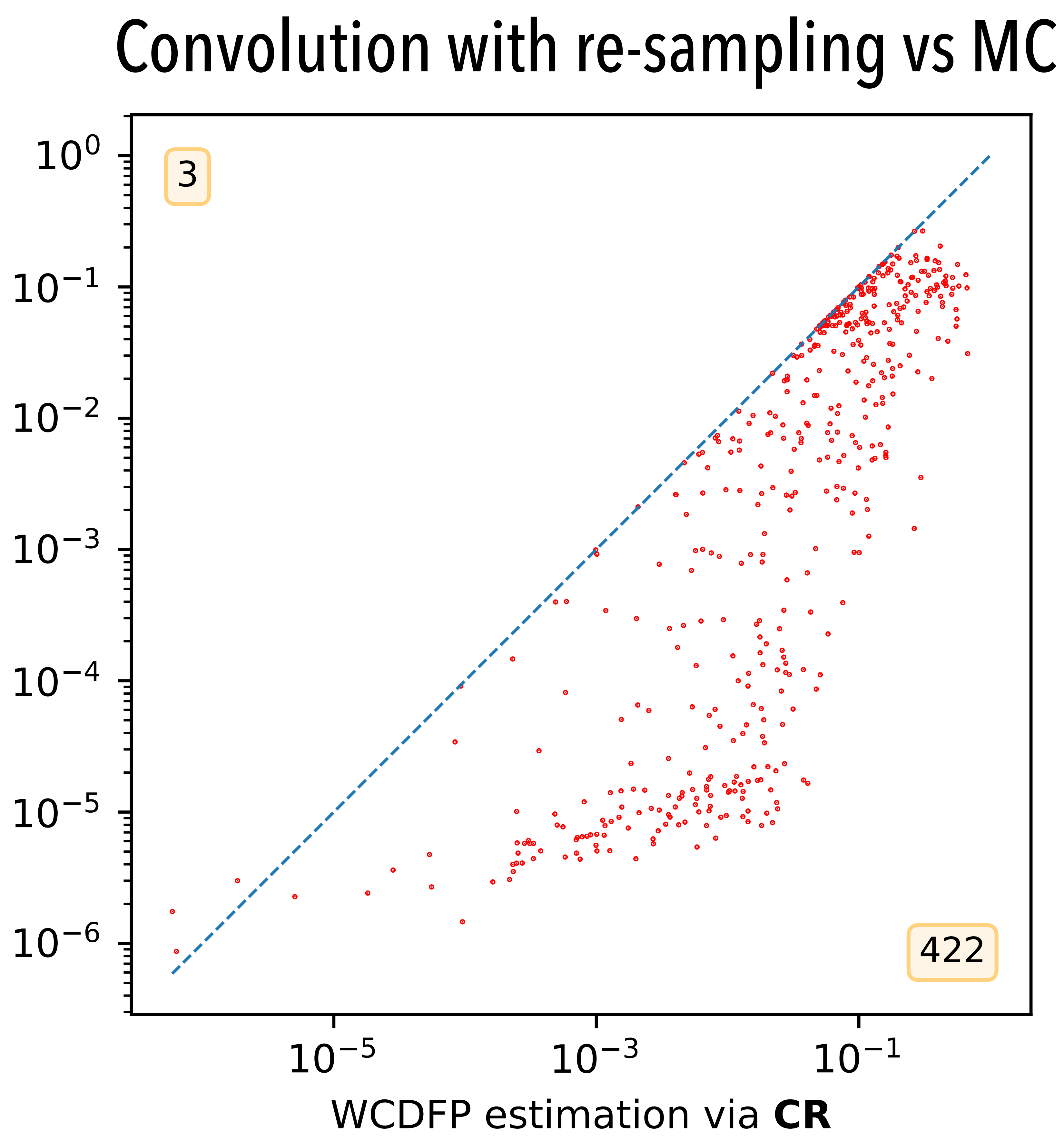
Convolution with re-sampling vs MC



- 500 task sets
- Shape: $\mathscr{C}_i = \begin{pmatrix} c & 3c \\ 0.95 & 0.05 \end{pmatrix}$
- Cardinality: $n \in \{5,10,\ldots 50\}$
- Utilization: $u \in \{0.75,0.8,\ldots,0.95\}$

EVALUATION (VARY COST DIFFERENCE)

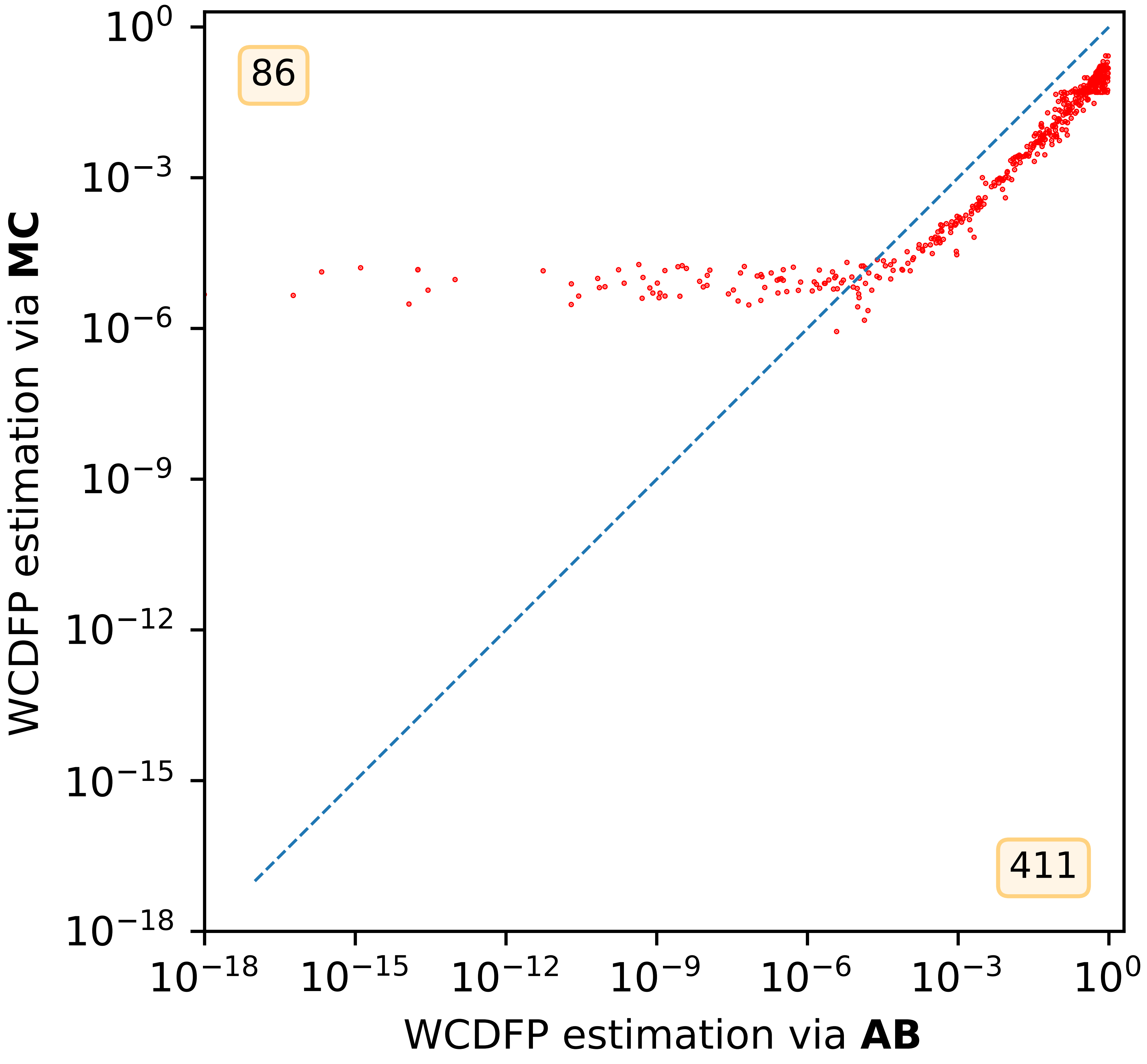




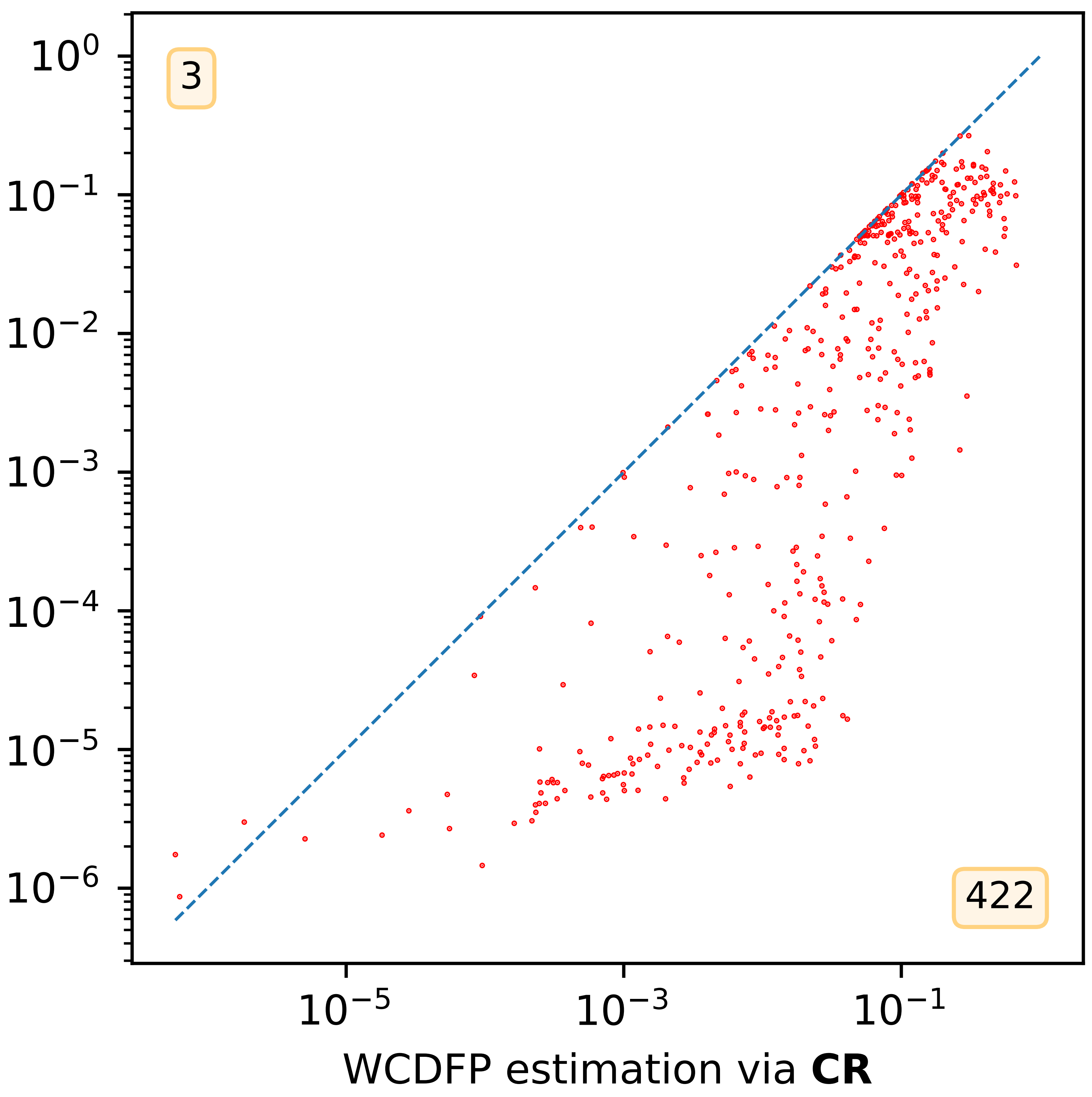
- 500 task sets
- Shape: $\mathscr{C}_i = \begin{pmatrix} c & 3c \\ 0.95 & 0.05 \end{pmatrix}$
- Cardinality: $n \in \{5, 10, \ldots 50\}$
- Utilization: $u \in \{0.75, 0.8, \ldots, 0.95\}$

EVALUATION (VARY COST DIFFERENCE)

Chernoff's inequality vs MC

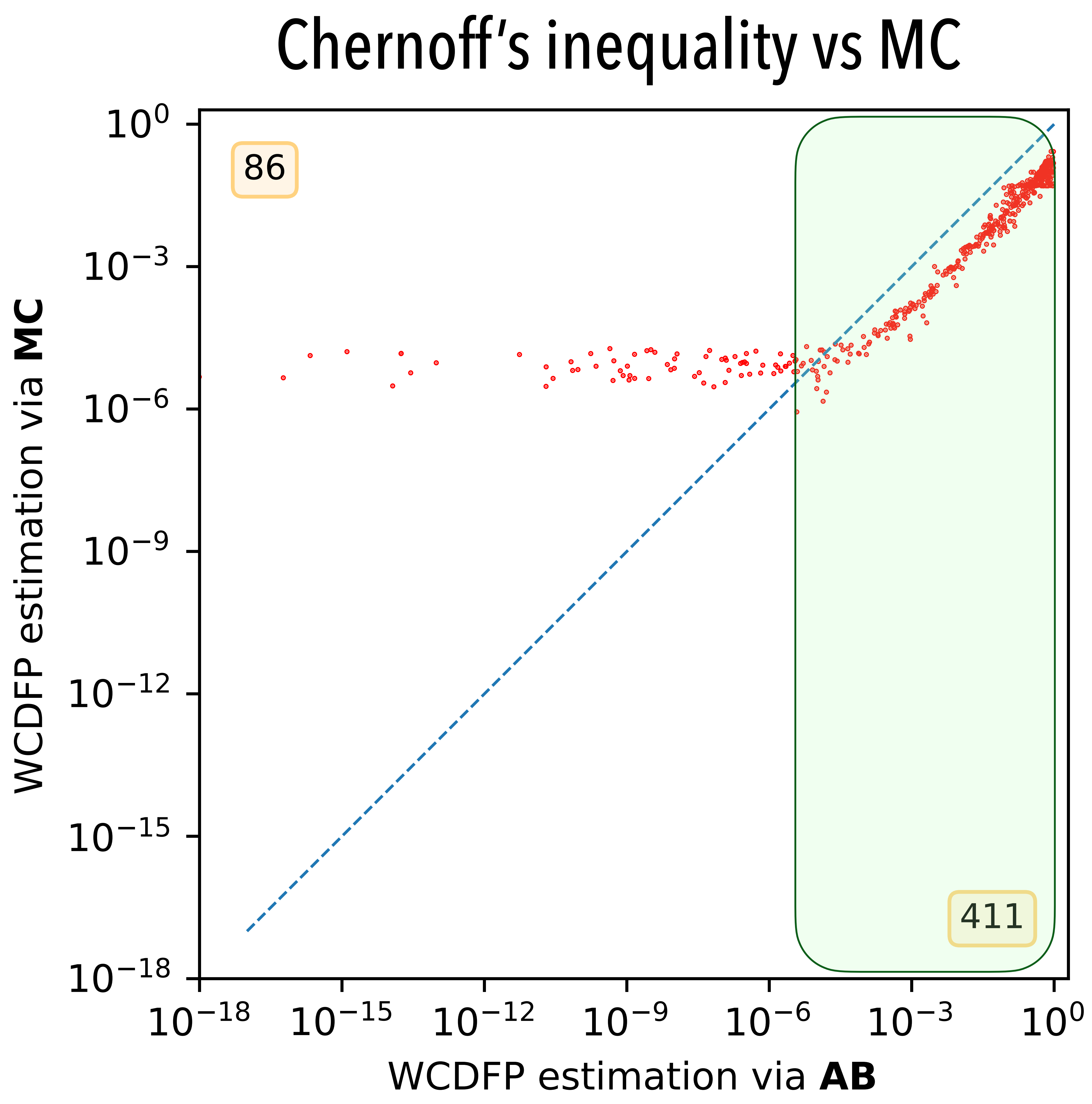


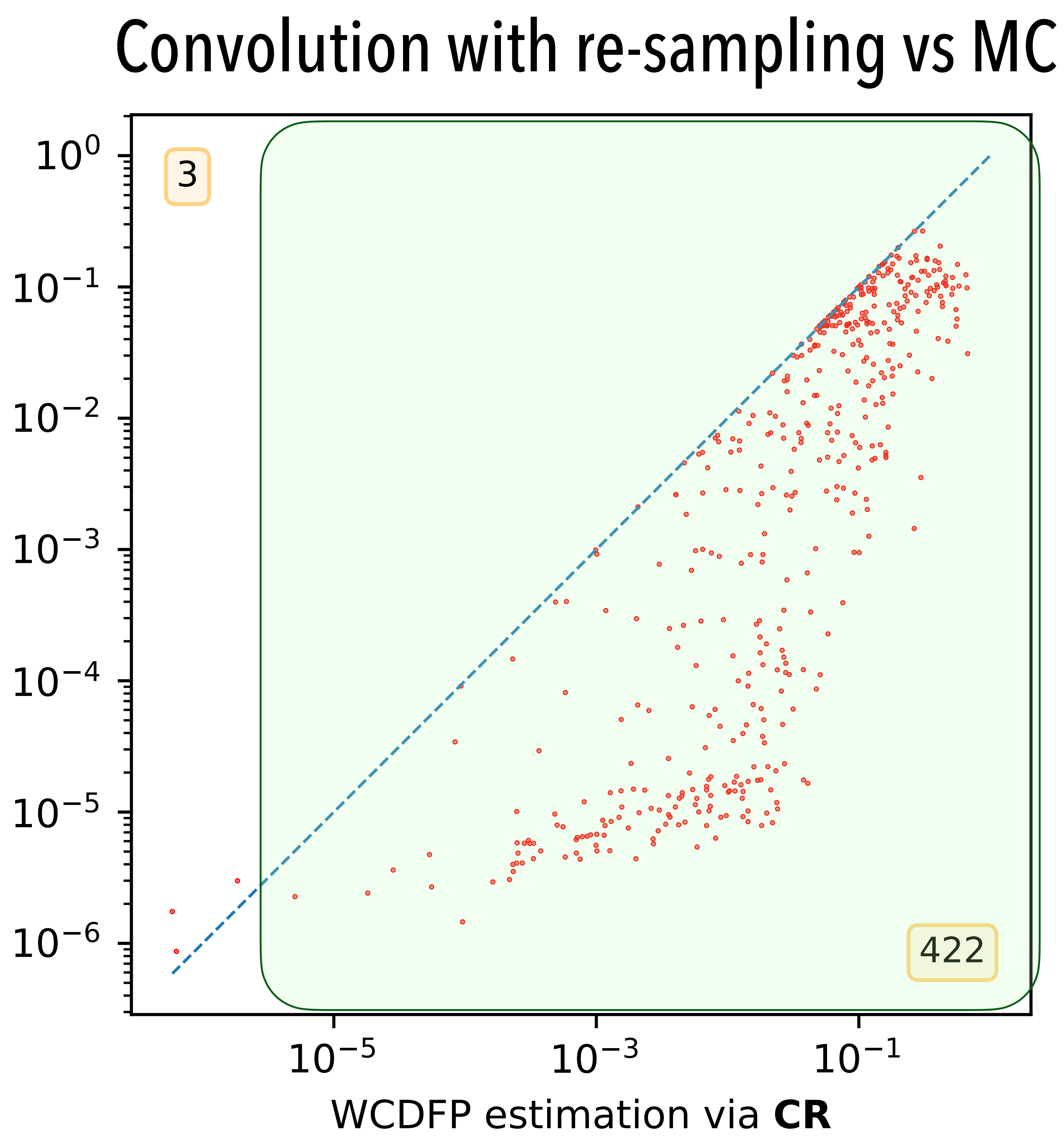
Convolution with re-sampling vs MC



- 500 task sets
- Shape: $\mathscr{C}_i = \begin{pmatrix} c & 3c \\ 0.95 & 0.05 \end{pmatrix}$
- Cardinality: $n \in \{5, 10, \ldots 50\}$
- Utilization: $u \in \{0.75, 0.8, \ldots, 0.95\}$

EVALUATION (VARY COST DIFFERENCE)

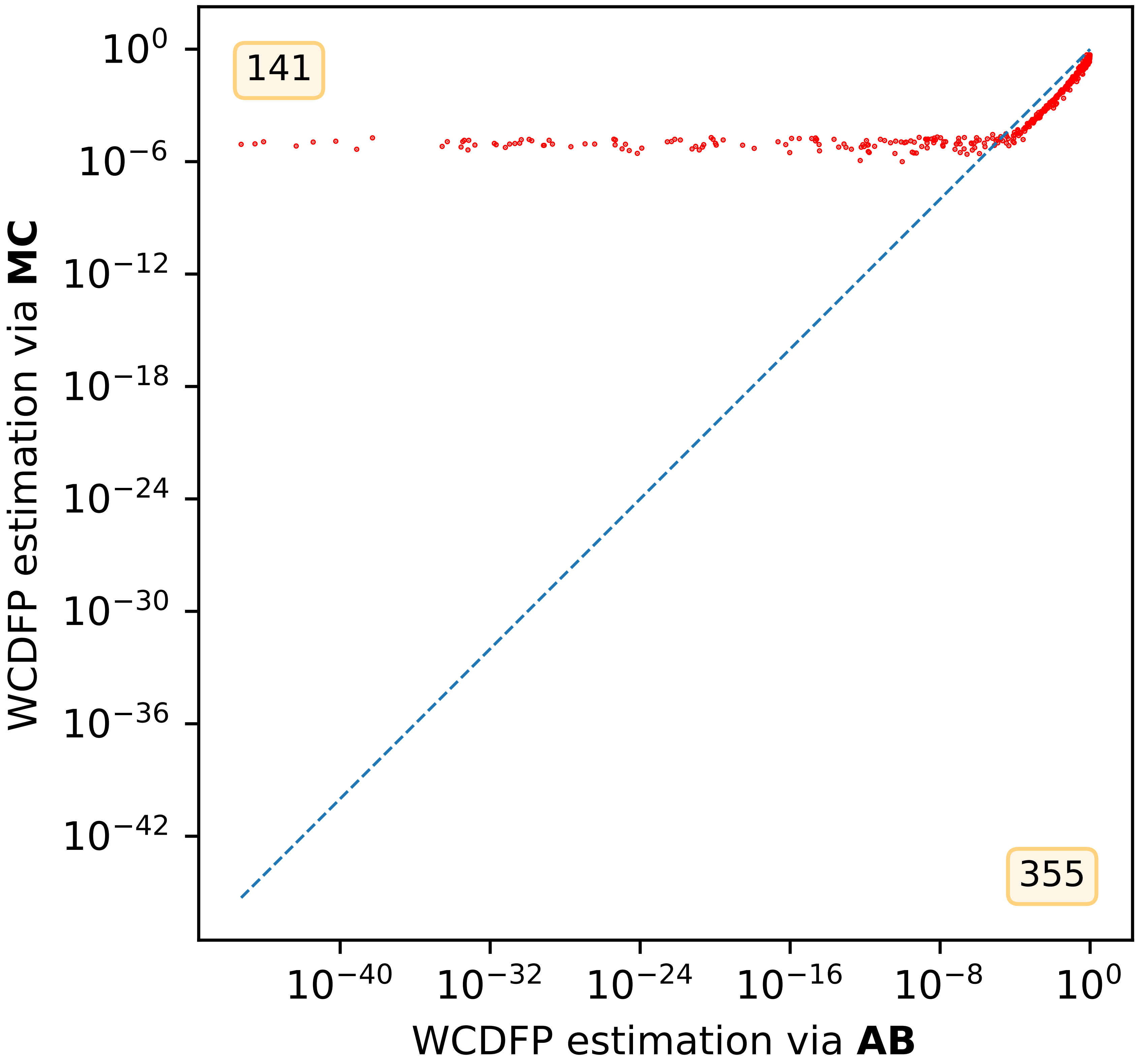




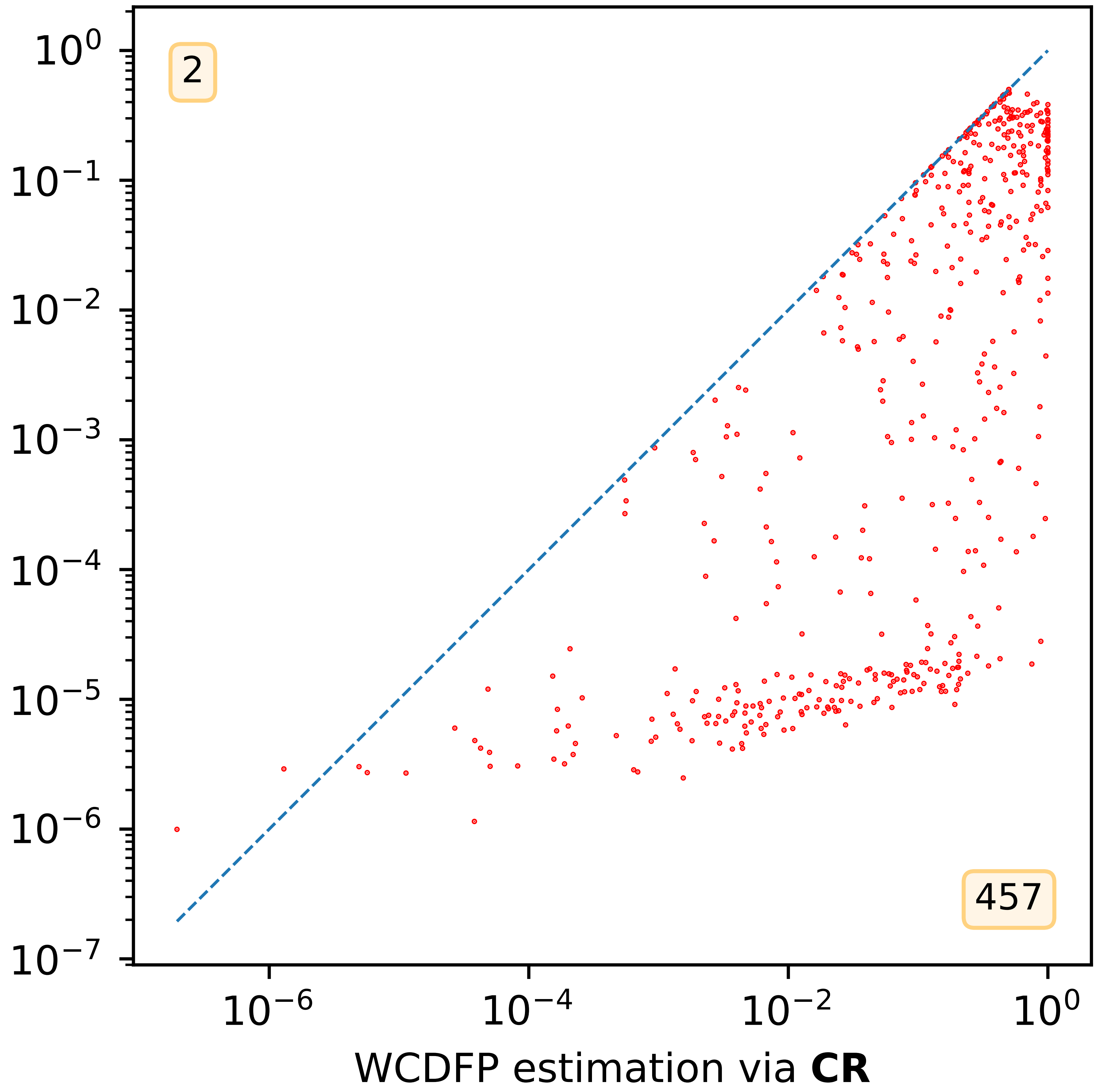
- 500 task sets
- Shape: $\mathscr{C}_i = \begin{pmatrix} c & 3c \\ 0.95 & 0.05 \end{pmatrix}$
- Cardinality: $n \in \{5, 10, \ldots 50\}$
- Utilization: $u \in \{0.75, 0.8, \ldots, 0.95\}$

EVALUATION (VARY MODE PROBABILITIES)

Chernoff's inequality vs MC



Convolution with re-sampling vs MC

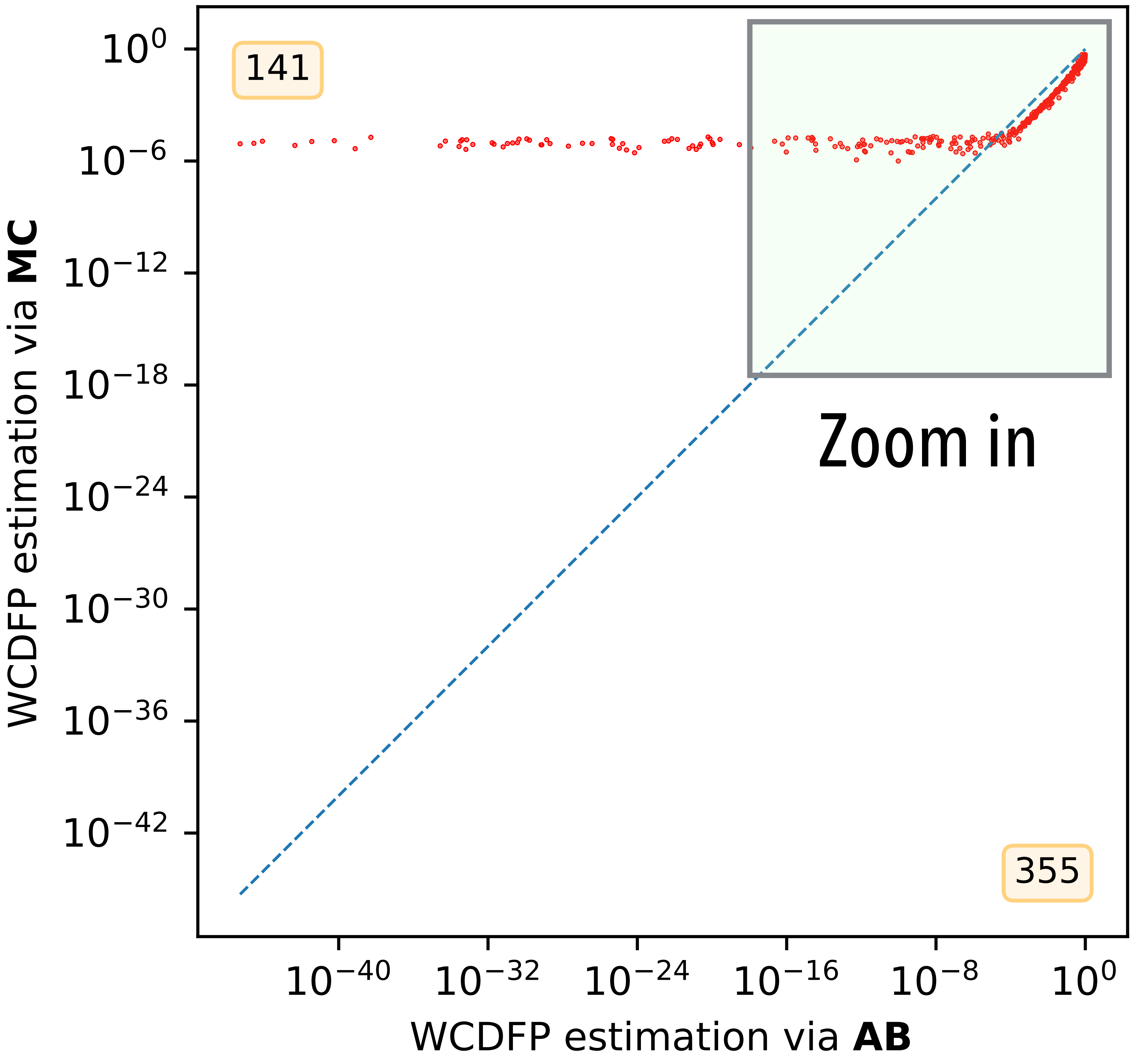


- ➔ 500 task sets
- ➔ Shape: $\mathscr{C}_i = \begin{pmatrix} c & 4c \\ 0.5 & 0.5 \end{pmatrix}$

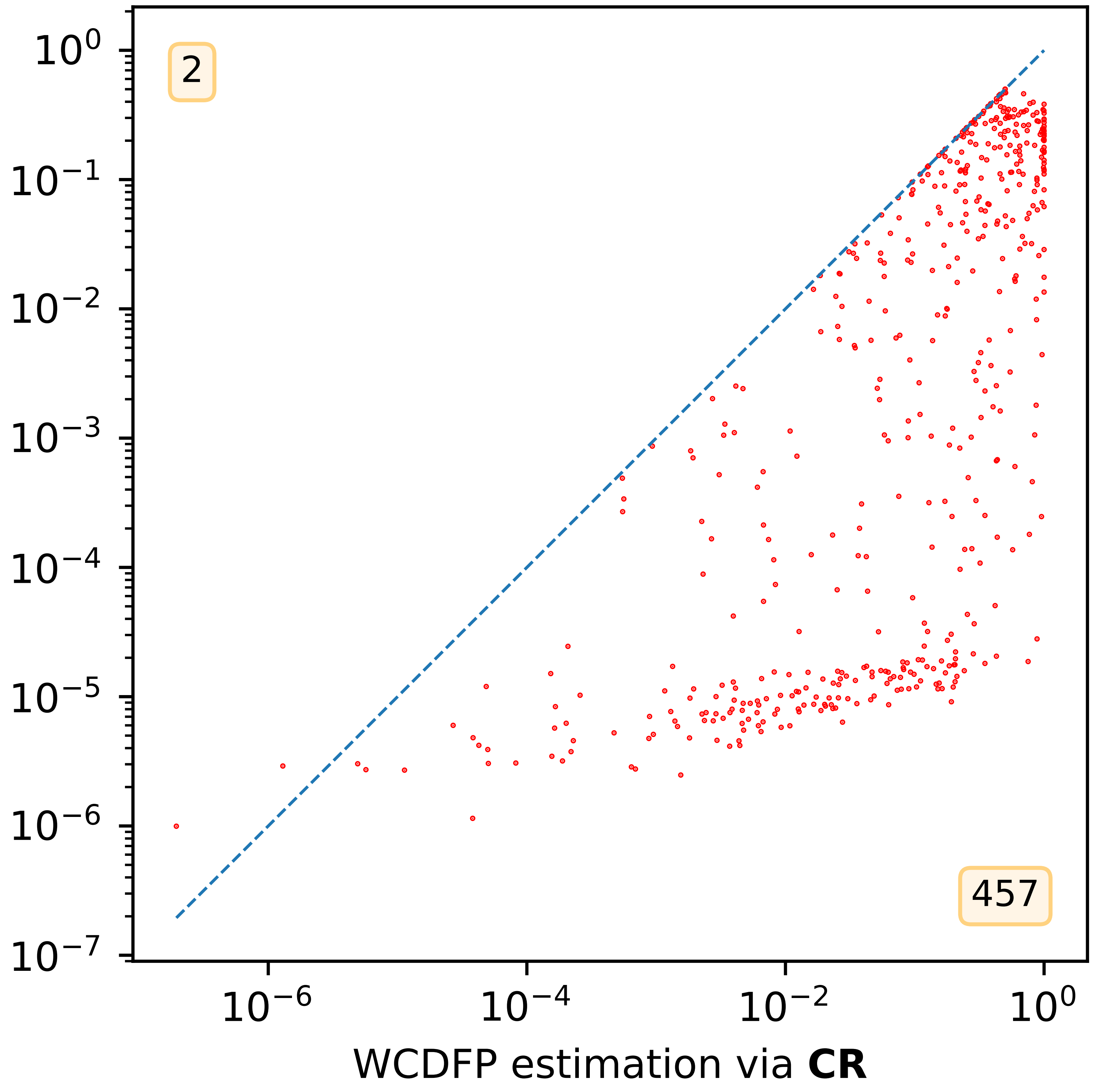
- ➔ Cardinality: $n \in \{5,10,\ldots 50\}$
- ➔ Utilization: $u \in \{0.75, 0.8, \ldots, 0.95\}$

EVALUATION (VARY MODE PROBABILITIES)

Chernoff's inequality vs MC

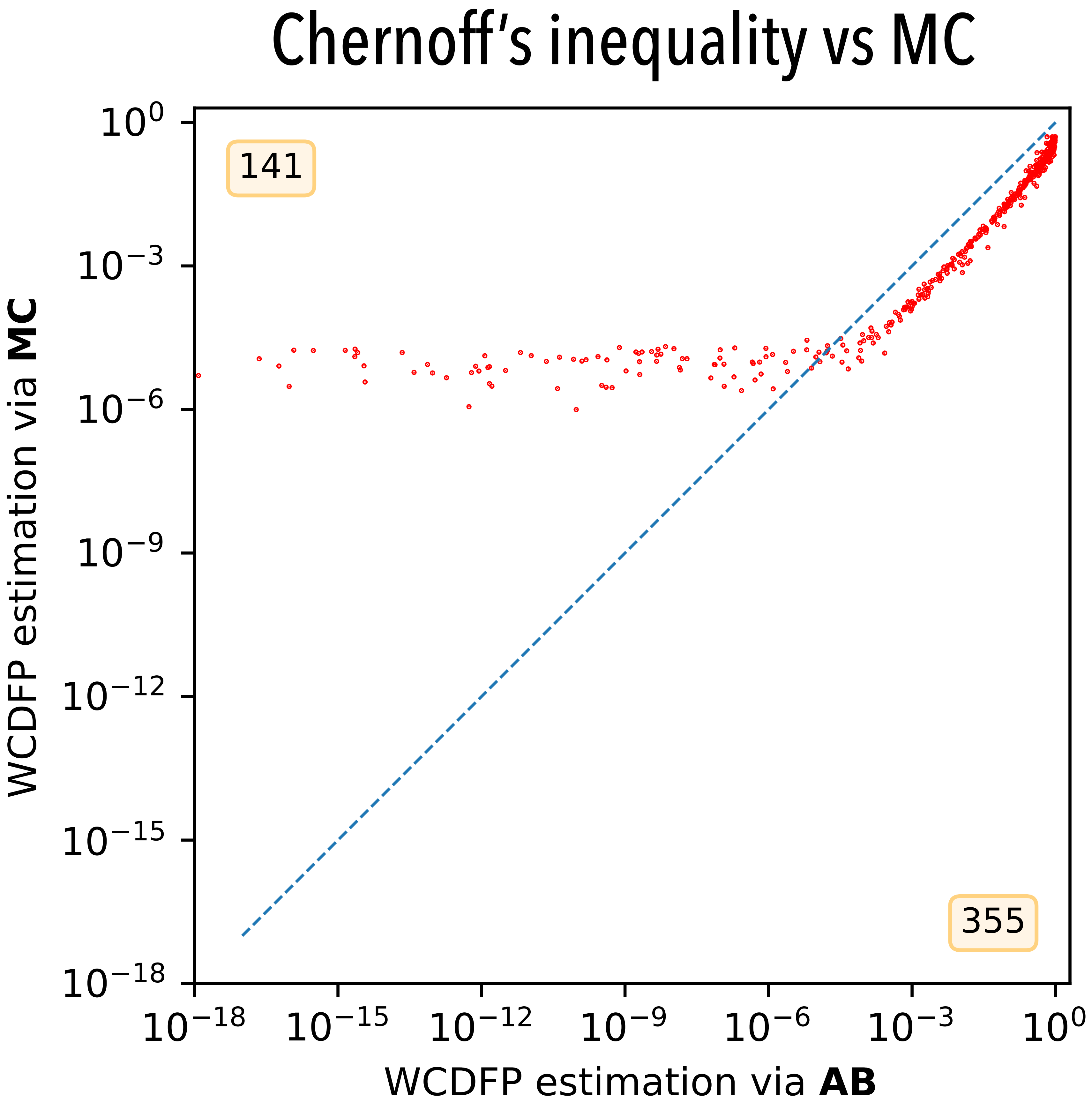


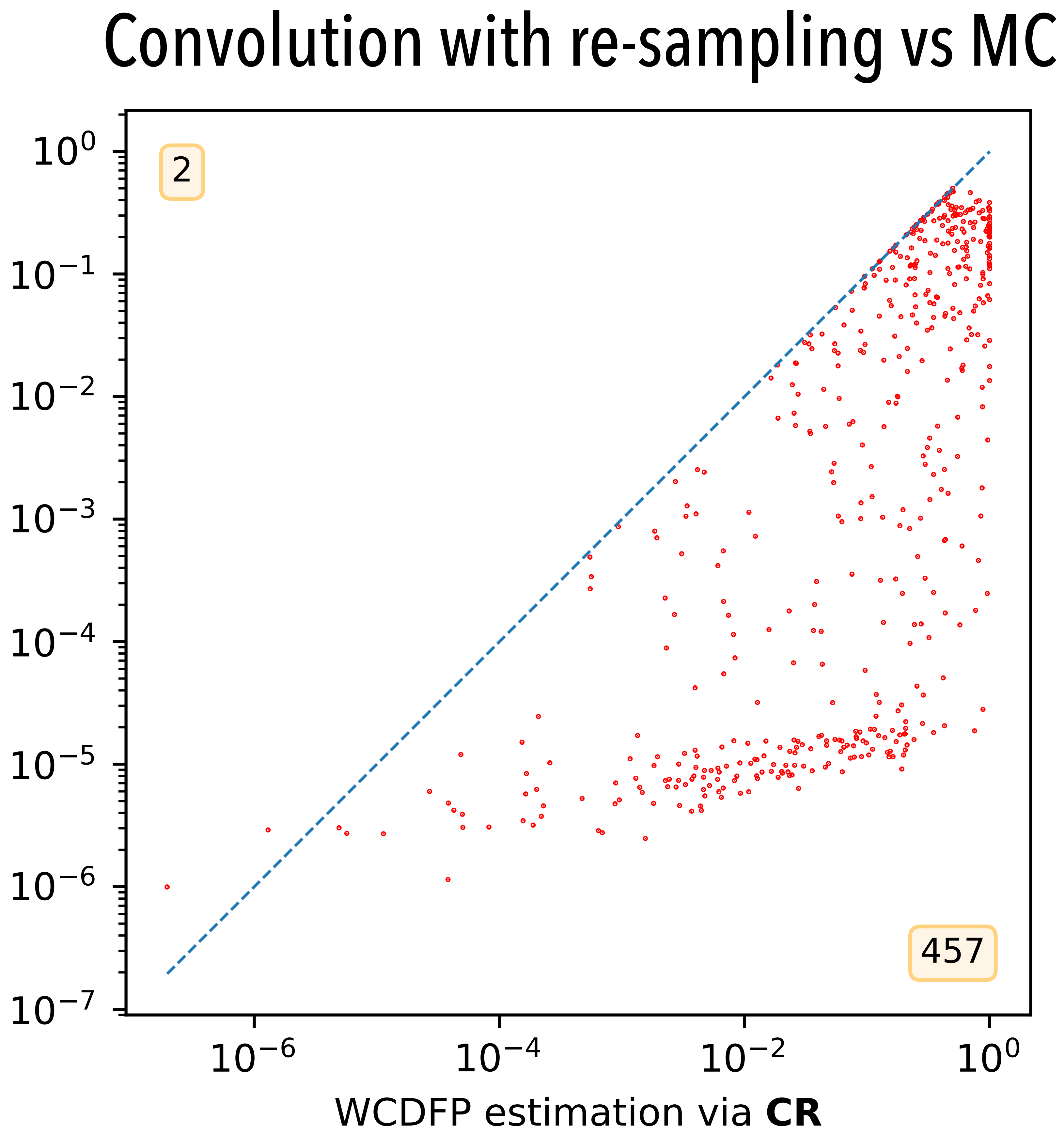
Convolution with re-sampling vs MC



- ➔ 500 task sets
- ➔ Shape: $\mathscr{C}_i = \begin{pmatrix} c & 4c \\ 0.5 & 0.5 \end{pmatrix}$
- ➔ Cardinality: $n \in \{5,10,\ldots 50\}$
- ➔ Utilization: $u \in \{0.75,0.8,\ldots,0.95\}$

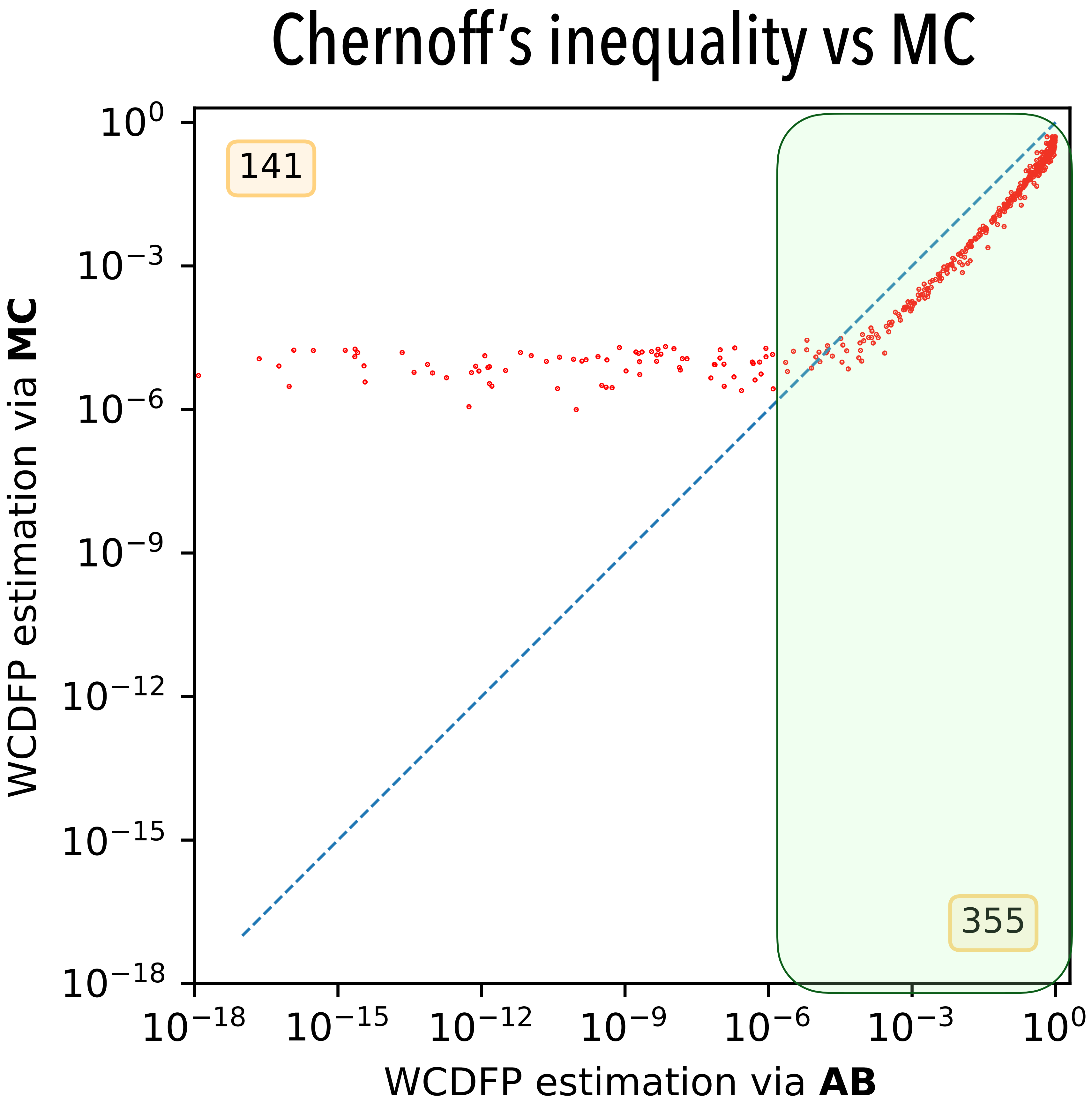
EVALUATION (VARY MODE PROBABILITIES)

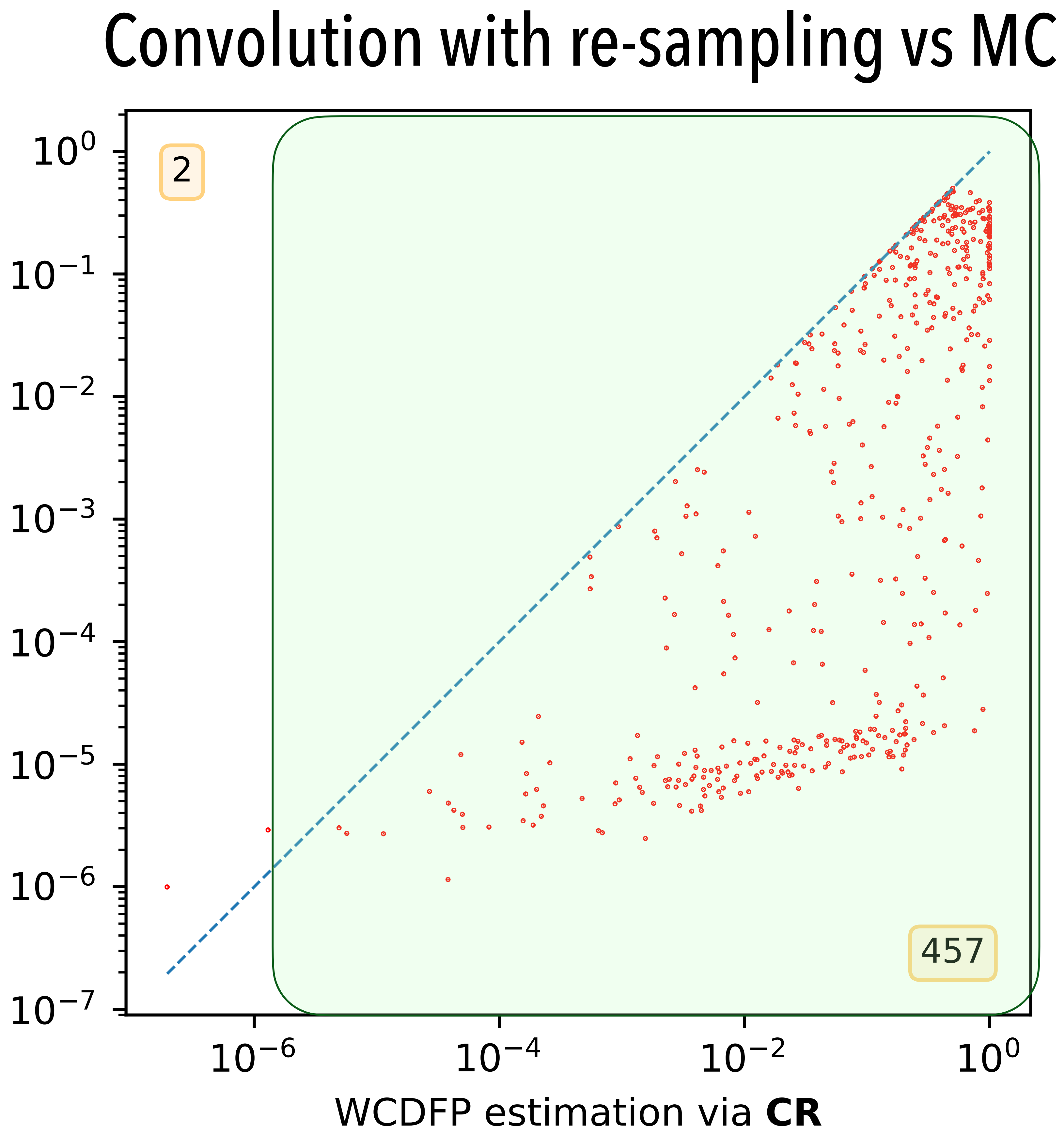




- ➔ 500 task sets
- ➔ Shape: $\mathscr{C}_i = \begin{pmatrix} c & 4c \\ 0.5 & 0.5 \end{pmatrix}$
- ➔ Cardinality: $n \in \{5,10,\ldots 50\}$
- ➔ Utilization: $u \in \{0.75,0.8,\ldots,0.95\}$

EVALUATION (VARY MODE PROBABILITIES)

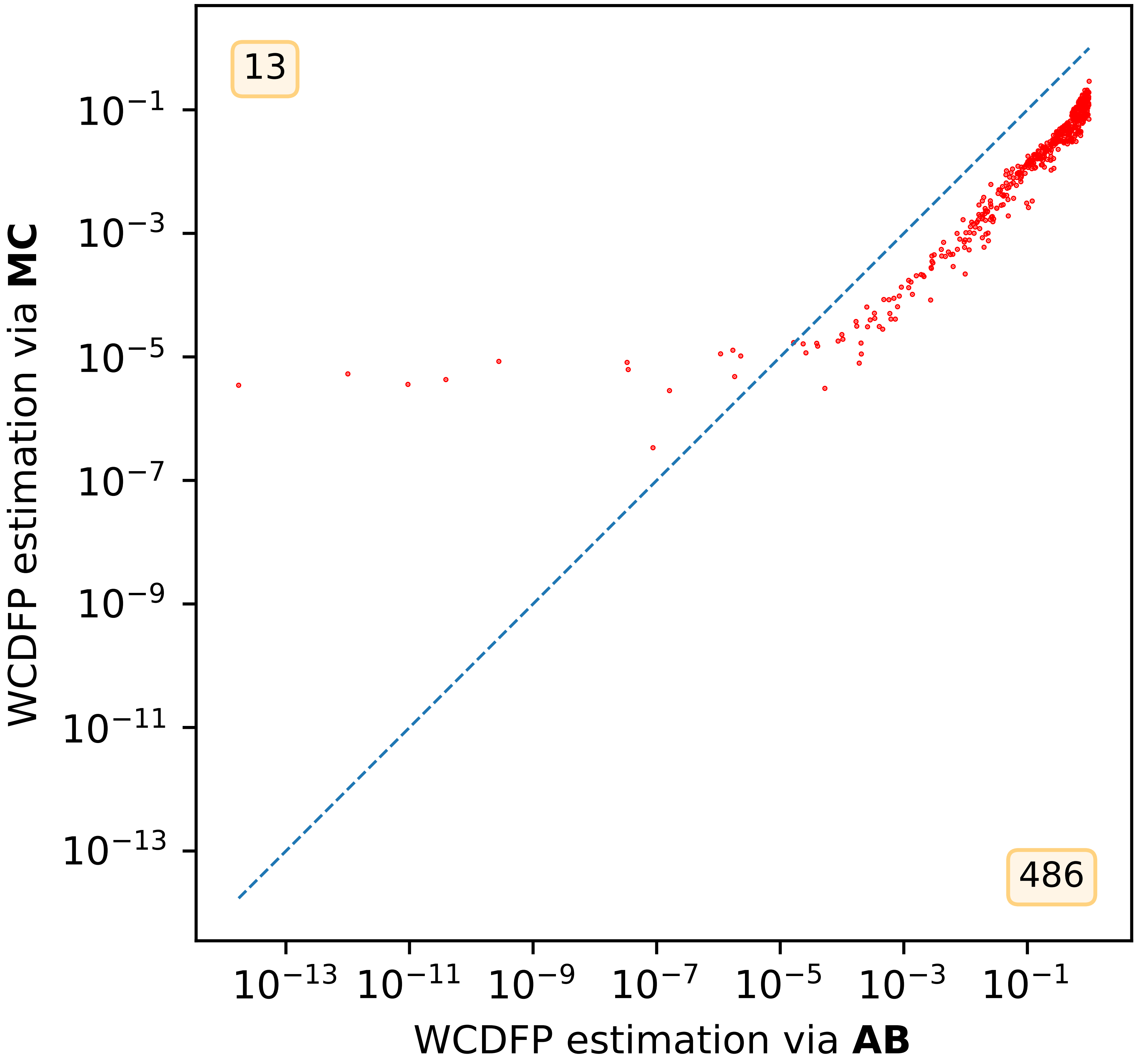




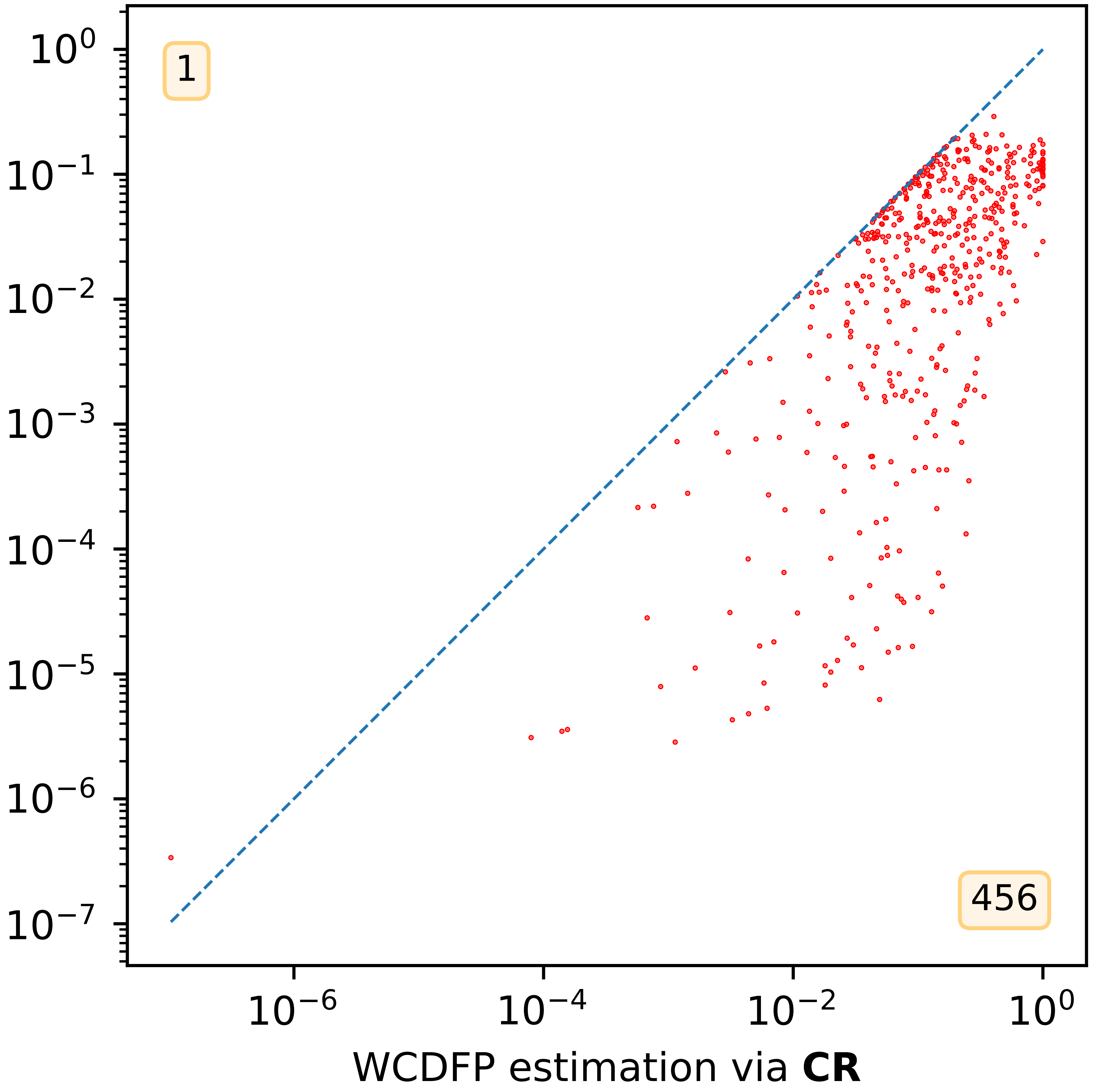
- ➔ 500 task sets
- ➔ Shape: $\mathscr{C}_i = \begin{pmatrix} c & 4c \\ 0.5 & 0.5 \end{pmatrix}$
- ➔ Cardinality: $n \in \{5{,}10{,}\ldots 50\}$
- ➔ Utilization: $u \in \{0.75{,}0.8{,}\ldots{,}0.95\}$

EVALUATION (VARY NUMBER OF MODES)

Chernoff's inequality vs MC



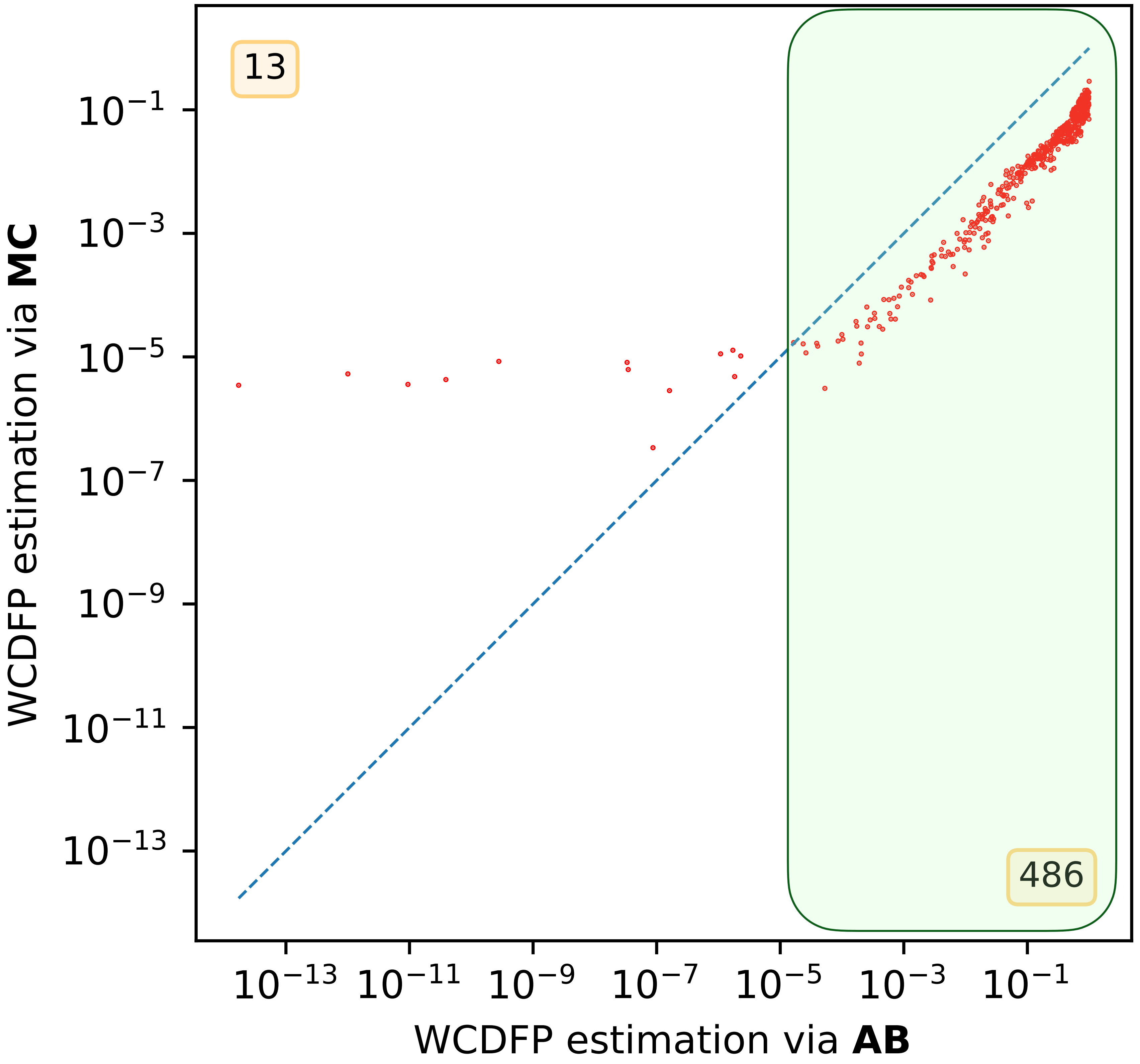
Convolution with re-sampling vs MC



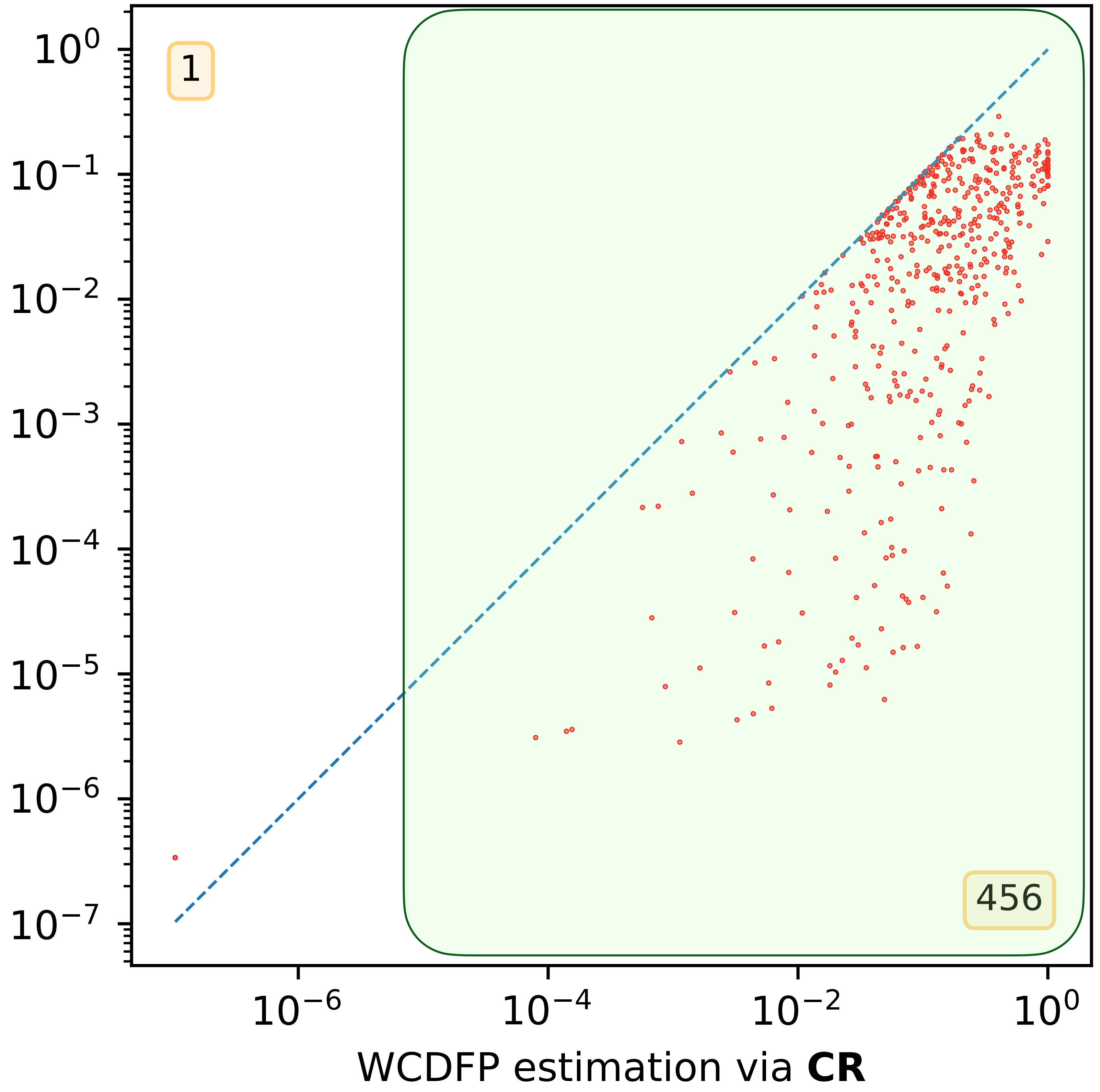
- 500 task sets
- Shape: $\mathscr{C}_i = \begin{pmatrix} c & 2c & 4c & 6c \\ 0.93 & 0.04 & 0.02 & 0.01 \end{pmatrix}$
- Cardinality: $n \in \{5,10,\ldots 50\}$
- Utilization: $u \in \{0.75,0.8,\ldots,0.95\}$

EVALUATION (VARY NUMBER OF MODES)

Chernoff's inequality vs MC



Convolution with re-sampling vs MC



- ➔ 500 task sets
- ➔ Shape: $\mathscr{C}_i = \begin{pmatrix} c & 2c & 4c & 6c \\ 0.93 & 0.04 & 0.02 & 0.01 \end{pmatrix}$
- ➔ Cardinality: $n \in \{5,10,\ldots 50\}$
- ➔ Utilization: $u \in \{0.75,0.8,\ldots,0.95\}$

CONCLUSION

Monte Carlo techniques can be used to great effect in RTS

➔ Try to apply Monte Carlo techniques to your favourite (unsolved) problem

Application of Monte Carlo techniques for WCDFP estimation:

➔ Less sensitive to input parameters than state-of-the-art

➔ Easy parameter tuning

➔ In most cases outperforms state-of-the-art approaches