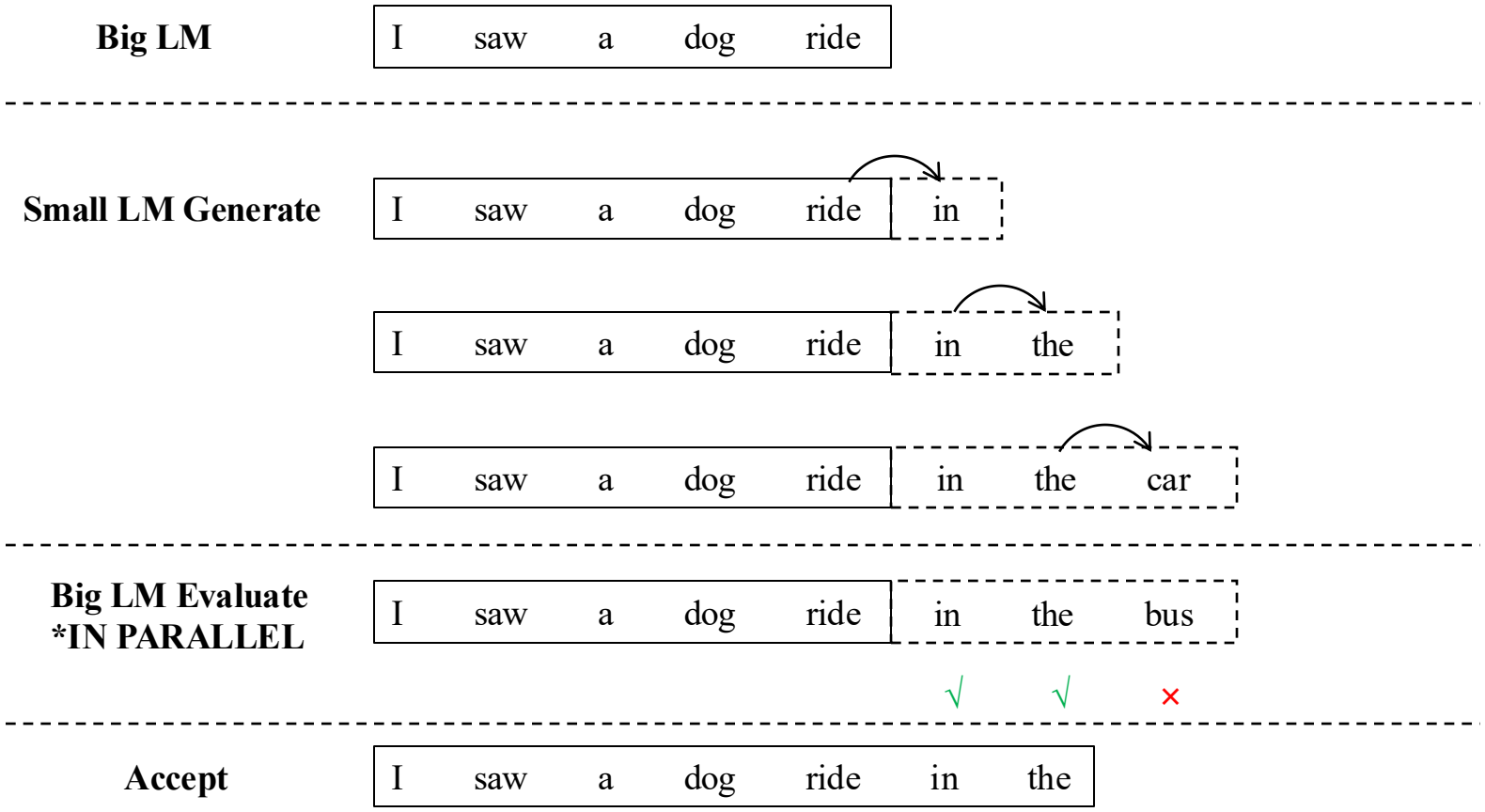
Traversal Verification for Speculative Tree Decoding

Yepeng Weng

Lenovo AI Technology Center, Lenovo

Speculative Decoding

- Speculative Decoding is a lossless **LLM acceleration method** which leverages a small, lightweight draft model to “speculate” the output of the original LLM, then use the original LLM to verify the speculated tokens in parallel.



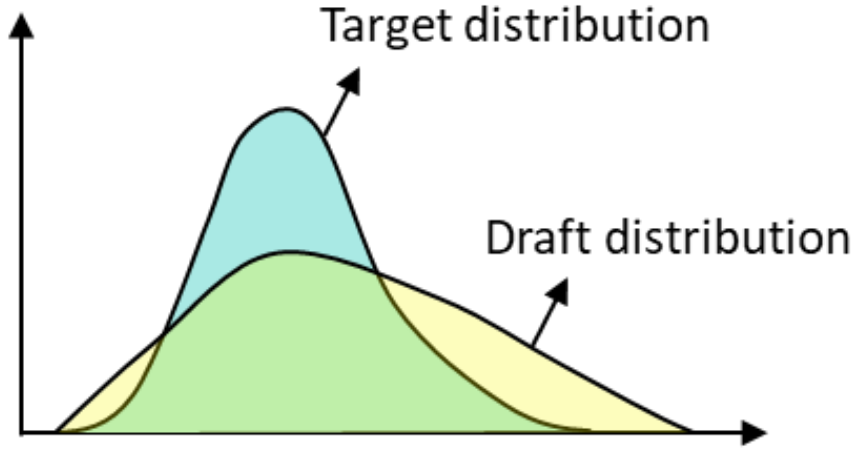
Verification method

- **When T=0**, verifying the candidates produced by the draft model is quite easy, as you only need to **check if the top-1 tokens of draft & target model are the same**.
- When T>0 (non-greedy), speculative decoding follows an acceptance mechanism called "**Rejection Sampling**", which is a little complex.
- The goal of rejection sampling is to **recover the target distribution from the draft distribution**. Given a target distribution and a draft distribution, the **acceptance lies in the overlap of these two distributions**. If rejected, you should resample a token from the residual distribution.

Algorithm 1 Single-token verification

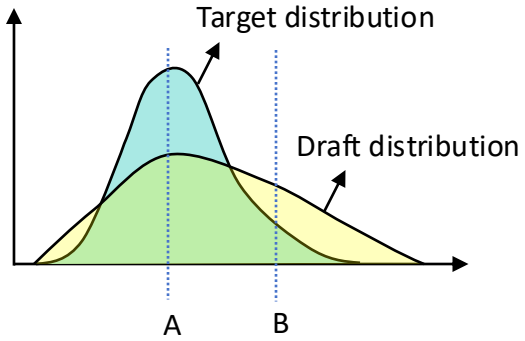
Input: Prefix X_0; draft token X; draft distribution $\mathcal{M}_s(\cdot|X_0)$; target distributions $\mathcal{M}_b(\cdot|X_0)$ and $\mathcal{M}_b(\cdot|X_0, X)$.

Sample $\eta \sim U(0, 1)$.
if $\eta < \frac{\mathcal{M}_b(X|X_0)}{\mathcal{M}_s(X|X_0)}$ **then**
  Sample Y from $\mathcal{M}_b(\cdot|X_0, X)$.
  Return: X, Y.
else
  Sample Y from norm$([\mathcal{M}_b - \mathcal{M}_s]_+)$.
  Return: Y.
end if



Token-level vs. Sequence-level

- Under standard speculative decoding, the acceptance rate is determined by the "**overlap**", therefore, given that you already sampled a token from draft distribution, there will be 2 cases:
  - If $p_{\text{target}} > p_{\text{draft}}$: Accept it (it's definitely in the overlap. e.g., sampled at A point)
  - If $p_{\text{target}} < p_{\text{draft}}$: Accept with probability $p_{\text{target}} / p_{\text{draft}}$ (an additional judge to determine if it's in the overlap. e.g., sampled at B point)



- Let's think about a toy example.
- If we have a chain with length=2, and the target and draft distributions are:
  - Target probability: 0.1(1st token) – 0.9(2nd token)
  - Draft probability:0.9(1st token) – 0.1(2nd token)
- According to speculative decoding, the acceptance rate should be:
  - Acceptance rate: 0.11(1st token) – 1.0(2nd token)
- However, if you consider the **sequence** probability instead of **per-token** probability, since the sequence-level probabilities are the same (both are 0.09), the entire sequence should be accepted immediately.

Traversal Verification

- In other words, **speculative decoding is not optimal due to per-token acceptance**.
- We rethink the foundations of speculative decoding algorithm, and proposed a traversal mechanism (post-order DFS) for speculative tree decoding.
- By **traversing the candidate tree** and using the **sequence-level probability,** we achieve longer acceptance length and higher speedup than vanilla speculative decoding.
- We theoretically proved the losslessness of Traversal Verification and its optimality in single-chain situations!

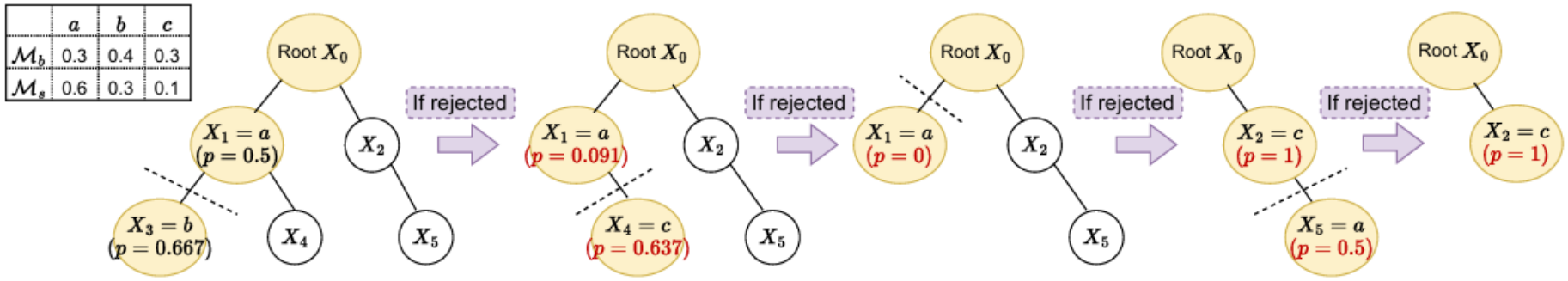


Figure 2: The traversal order of verifying a sampling tree.

Another perspective

- Recursive Rejection Sampling without Replacement (RRSw) is a lossless probability modification method for speculative tree decoding, which recursively redistributes the residual probability to other candidates after rejections.
- However, the probabilities of RRSw only "flow" within the same layer of a tree.
- Traversal Verification can be regarded as a **sequence-level RRSw**. As shown in the figure, we first transform the original decoding tree on the left into the right one, and then utilize the classic RRSw algorithm to derive the correct probability transition formulas.

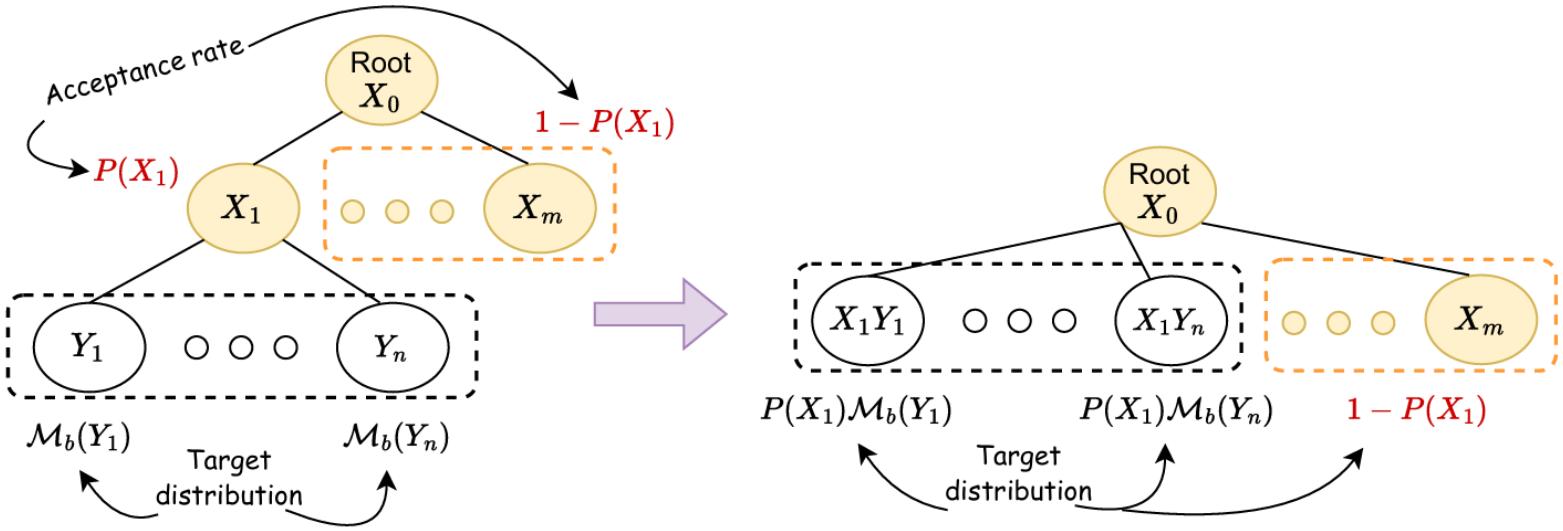


Figure 4: The sequence-level RRSw for two-layers decoding tree.

Experiments – Overall Effectiveness

- We present the acceptance lengths and throughput of two combinations of draft and target model, namely Llama3.2-1B-Instruct with Llama3.1-8B-Instruct and Llama-68M with Llama2-7B. For chain and binary tree, we set the depth at 5.
- As can be observed from the results, compared with token-level verification, Traversal Verification achieves an average improvement in acceptance length of 2.2% to 5.7% across different tasks, tree architectures, and combinations of draft and target models. The performance gains from Traversal Verification exhibit variability depending on the specific configurations of draft and target models.
- The generation speed without speculative decoding for Llama3.1-8B-Instruct is 34.5 token/s and for Llama2-7B is 37.3 token/s, and the speedup ratio can be calculated accordingly.

Table 2: Acceptance length and throughput on Llama3.2-1B-Instruct with Llama3.1-8B-Instruct.

Llama3.2-1B-Instruct (draft) & Llama3.1-8B-Instruct (target) Temperature=1									
	Chain			Binary Tree			EAGLE Sparse Tree		
Tasks	Tok.V	Tra.V	Δ	Tok.V	Tra.V	Δ	Tok.V	Tra.V	Δ
Multi-turn	$3.95_{\pm 0.03}$	$4.09_{\pm 0.03}$	3.5%	$4.64_{\pm 0.05}$	$4.76_{\pm 0.04}$	2.6%	$4.53_{\pm 0.02}$	$4.67_{\pm 0.02}$	3.1%
Translation	$3.50_{\pm 0.02}$	$3.53_{\pm 0.04}$	1.0%	$4.28_{\pm 0.02}$	$4.43_{\pm 0.03}$	3.4%	$4.16_{\pm 0.04}$	$4.27_{\pm 0.03}$	2.6%
Sum.	$3.66_{\pm 0.02}$	$3.76_{\pm 0.03}$	2.6%	$4.51_{\pm 0.02}$	$4.64_{\pm 0.02}$	2.7%	$4.32_{\pm 0.03}$	$4.46_{\pm 0.03}$	3.1%
QA	$3.51_{\pm 0.02}$	$3.68_{\pm 0.03}$	4.7%	$4.32_{\pm 0.05}$	$4.40_{\pm 0.04}$	2.0%	$4.19_{\pm 0.05}$	$4.31_{\pm 0.06}$	2.9%
Math	$4.61_{\pm 0.05}$	$4.70_{\pm 0.03}$	1.8%	$5.37_{\pm 0.03}$	$5.39_{\pm 0.05}$	0.4%	$5.13_{\pm 0.01}$	$5.21_{\pm 0.02}$	1.5%
RAG	$4.05_{\pm 0.04}$	$4.17_{\pm 0.05}$	3.1%	$4.63_{\pm 0.02}$	$4.76_{\pm 0.06}$	2.8%	$4.60_{\pm 0.03}$	$4.68_{\pm 0.04}$	1.7%
Avg. Accept.	$3.88_{\pm 0.02}$	$3.99_{\pm 0.01}$	2.8%	$4.63_{\pm 0.03}$	$4.73_{\pm 0.01}$	2.2%	$4.49_{\pm 0.02}$	$4.60_{\pm 0.02}$	2.4%
Avg. Token/s	$51.2_{\pm 1.2}$	$52.5_{\pm 1.1}$	2.5%	$54.0_{\pm 0.6}$	$54.9_{\pm 1.2}$	1.7%	$57.3_{\pm 1.3}$	$58.5_{\pm 0.8}$	2.1%

Table 3: Acceptance length and throughput on Llama-68M with Llama2-7B.

Llama-68M (draft) & Llama2-7B (target) Temperature=1									
	Chain			Binary Tree			EAGLE Sparse Tree		
Tasks	Tok.V	Tra.V	Δ	Tok.V	Tra.V	Δ	Tok.V	Tra.V	Δ
Multi-turn	$2.05_{\pm 0.05}$	$2.16_{\pm 0.03}$	5.5%	$2.47_{\pm 0.01}$	$2.59_{\pm 0.01}$	4.7%	$2.55_{\pm 0.02}$	$2.70_{\pm 0.02}$	5.6%
Translation	$1.97_{\pm 0.05}$	$2.10_{\pm 0.05}$	6.3%	$2.38_{\pm 0.01}$	$2.43_{\pm 0.03}$	2.1%	$2.49_{\pm 0.01}$	$2.51_{\pm 0.03}$	0.9%
Sum.	$1.77_{\pm 0.04}$	$1.86_{\pm 0.05}$	4.9%	$2.14_{\pm 0.01}$	$2.27_{\pm 0.03}$	5.8%	$2.25_{\pm 0.02}$	$2.36_{\pm 0.02}$	4.7%
QA	$2.07_{\pm 0.01}$	$2.19_{\pm 0.02}$	5.6%	$2.59_{\pm 0.05}$	$2.71_{\pm 0.01}$	4.8%	$2.63_{\pm 0.02}$	$2.69_{\pm 0.02}$	2.2%
Math	$2.01_{\pm 0.05}$	$2.15_{\pm 0.04}$	7.0%	$2.49_{\pm 0.05}$	$2.67_{\pm 0.06}$	7.0%	$2.57_{\pm 0.02}$	$2.72_{\pm 0.01}$	6.0%
RAG	$2.09_{\pm 0.05}$	$2.19_{\pm 0.03}$	4.8%	$2.56_{\pm 0.05}$	$2.69_{\pm 0.05}$	5.0%	$2.63_{\pm 0.02}$	$2.71_{\pm 0.06}$	3.2%
Avg. Accept.	$1.99_{\pm 0.01}$	$2.10_{\pm 0.01}$	5.7%	$2.44_{\pm 0.03}$	$2.56_{\pm 0.01}$	4.9%	$2.52_{\pm 0.01}$	$2.62_{\pm 0.01}$	3.8%
Avg. Token/s	$58.0_{\pm 0.7}$	$60.8_{\pm 0.8}$	4.8%	$59.4_{\pm 0.8}$	$61.6_{\pm 0.6}$	3.7%	$69.1_{\pm 0.9}$	$71.2_{\pm 1.0}$	3.0%

Experiments – Impact of Chain Depth and Tree Size

- Since Traversal Verification considers the joint probability of the entire sequence, it is intuitive that the performance improvement will become more pronounced as the tree size and depth increase. To illustrate these effects, we perform experiments across varying chain depths and tree sizes, as shown in the following figure.
- The advantage of Traversal Verification grows progressively with increasing chain depth and tree size. In specialized scenarios (e.g., model offloading) where large tree sizes are permissible, Traversal Verification is expected to demonstrate even greater performance gains.

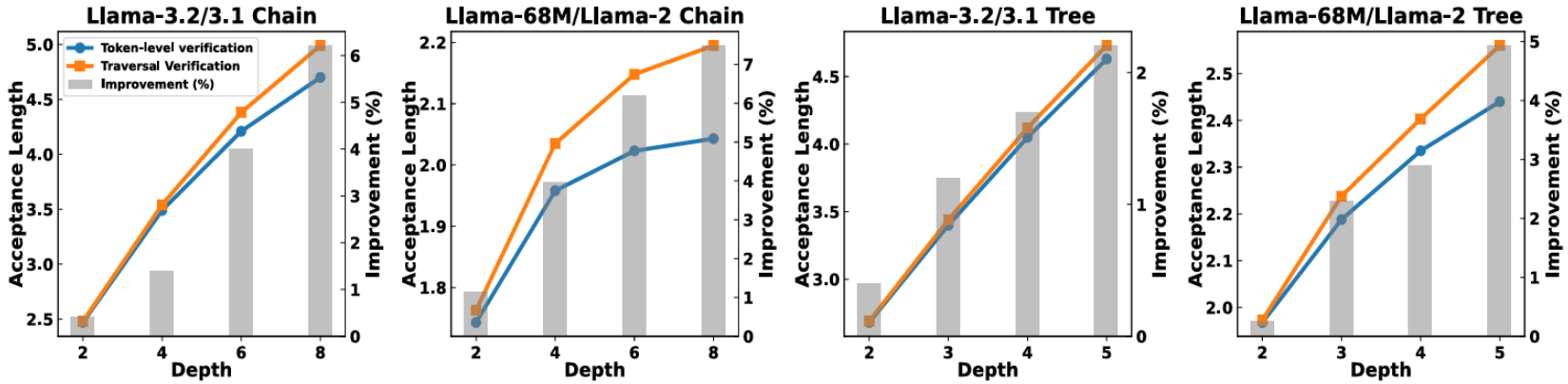


Figure 3: Acceptance lengths and improvements under different chain depths and tree sizes.

Experiments – Impact of Temperature

- Intuitively, as the temperature decreases (i.e., the probability distribution becomes more concentrated), the performance gap between token-level verification and Traversal Verification narrows.
- Conversely, at higher temperatures, Traversal Verification demonstrates more pronounced advantages.
- We use Llama3.2-1B-Instruct and Llama3.1-8B-Instruct as the draft and target models, respectively.

Table 4: Acceptance lengths under different temperature.

	Chain			Binary Tree			EAGLE Sparse Tree		
Temp.	Tok.V	Tra.V	Δ	Tok.V	Tra.V	Δ	Tok.V	Tra.V	Δ
0.2	$4.16_{\pm 0.01}$	$4.20_{\pm 0.01}$	1.0%	$5.01_{\pm 0.02}$	$5.07_{\pm 0.02}$	1.2%	$4.77_{\pm 0.03}$	$4.84_{\pm 0.01}$	1.5%
0.4	$4.14_{\pm 0.02}$	$4.20_{\pm 0.02}$	1.4%	$5.00_{\pm 0.01}$	$5.06_{\pm 0.01}$	1.2%	$4.76_{\pm 0.02}$	$4.83_{\pm 0.02}$	1.5%
0.6	$4.11_{\pm 0.02}$	$4.17_{\pm 0.03}$	1.5%	$4.92_{\pm 0.03}$	$5.00_{\pm 0.01}$	1.5%	$4.71_{\pm 0.01}$	$4.78_{\pm 0.01}$	1.5%
0.8	$4.02_{\pm 0.02}$	$4.11_{\pm 0.01}$	2.2%	$4.81_{\pm 0.02}$	$4.90_{\pm 0.02}$	1.7%	$4.64_{\pm 0.02}$	$4.72_{\pm 0.01}$	1.7%
1.0	$3.88_{\pm 0.02}$	$3.99_{\pm 0.01}$	2.8%	$4.63_{\pm 0.03}$	$4.73_{\pm 0.01}$	2.2%	$4.49_{\pm 0.02}$	$4.60_{\pm 0.02}$	2.4%

Thanks!