清華大學电子工程系
Department of Electronic Engineering, Tsinghua University

NEURAL INFORMATION PROCESSING SYSTEMS

MoESD: Unveil Speculative Decoding’s Potential for Accelerating Sparse MoE

Zongle Huang, Lei Zhu, Zongyuan Zhan, Ting Hu, Weikai Mao, Xianzhi Yu, Yongpan Liu, Tianyu Zhang

39th Conference on Neural Information Processing Systems (NeurIPS 2025)

Introduction

- Speculative decoding (SD) is a lossless method for LLM acceleration
  - Using small draft models to **generate** tokens rapidly
  - Using the original large model (target model) to **verify** them
- However, it has been believed less effective for MoE
  - For *Mixtral 8x7B Instruct-v0.1* **(MoE)**: **1.5x** speedup
  - For *Llama / Vicuna* **(dense models)** :**~3x** speedup
  - (Source from Eagle paper)
- What we want to exhibit in this work:
  - Is it possible that SD is also effective for MoE? (Surprisingly yes or even better!)
  - How to evaluate SD more comprehensively?

Formulation of SD and Introduction of Target Efficiency

$$T_{SD} = R \times (T_{propose} + T_{verify} + T_{reject}) = R \times \left(\gamma \cdot T_D(B,1) + T_T(B,\gamma) + T_{reject}\right)$$

$$Speedup = \frac{T_{AR}}{T_{SD}} = \frac{S \cdot T_T(B,1)}{R \cdot \left(\gamma \cdot T_D(B,1) + T_T(B,\gamma) + T_{reject}\right)}$$

$$= \boxed{\frac{S}{R}} \cdot \frac{1}{\gamma \cdot \frac{T_D(B,1)}{T_T(B,1)} + \boxed{\frac{T_T(B,\gamma)}{T_T(B,1)}} + \frac{T_{reject}}{T_T(B,1)}}$$

reciprocal

- Besides **acceptance rate**, **target efficiency is also a critical factor**
- When target efficiency **gets low:**
  - Compute-boundness.
  - The extra memory loads.

- R: # rounds of speculation for sequence with given length
- γ: # draft tokens per speculation
- $T_{T/D}(b, s)$: the time for once forwarding of the target / draft model, b for batch size and s for the number of tokens to process.

Specialization to MoE

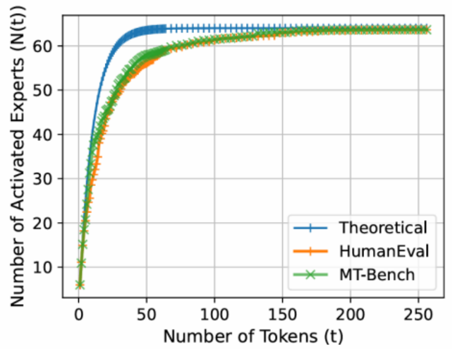
$$N = \sum_i \mathbb{E}[X_i] = \sum_i Pr(X_i) = E \cdot Pr(X)$$

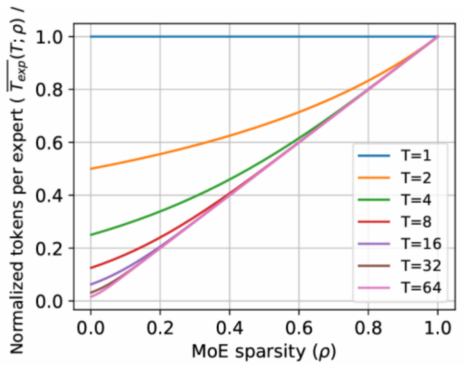
$$Pr(X) = 1 - Pr(\text{ None of the } t \text{ tokens activates the expert }) = 1 - (\frac{E-K}{E})^t$$

$$N(t) = E \cdot \left(1 - (\frac{E-K}{E})^t\right)$$

$$N(T_{thres}) = E \cdot \left(1 - (1-\rho)^{T_{thres}}\right) \geq \tau E \quad \Rightarrow \quad T_{thres} = \lceil \log_{(1-\rho)}(1-\tau) \rceil$$

$$\overline{T_{exp}}(t;\rho) = \frac{t \cdot K}{N} = \frac{t \cdot (\rho E)}{E \cdot \left(1 - (1-\rho)^t\right)} = \frac{\rho t}{1 - (1-\rho)^t}$$





- Revisiting factors that impact target efficiency
  - Extra memory loads: $t > N(T_{thres})$
  - Compute-boundness: $\overline{T_{exp}}(t;\rho)$, better than dense model!
  - Conclusion: in moderate batch size, SD would be **more effective** for **sparser MoEs**.

Modeling Method

Algorithm 1 The Modeling of SD Speedup and Corresponding Fitting Method

Measurement Input: A total of m measurements denoted as $\mathbf{M}$. Each $\mathbf{M}_i, i = 1, 2, ..., m$ contains the attributes including batch size B, draft length γ, number of activated experts per token K, total number of experts E, the ratio of accepted token counts to the maximal possible accepted tokens σ, *Speedup* for the actual speedup achieved.

Output: The optimal fitting parameter *params**.

def ComputeSpeedup($params$, B, γ, K, E, σ): ▷ Compute the SD Speedup

  $bias, k_1, k_2, k_3, draft\_bias, draft\_k, reject\_bias, reject\_k, \lambda, s = params$ ▷ Unpack parameters

  $N_{ar} = E \cdot (1 - ((E-K)/E)^B), \ \ T_{ar} = B \cdot K / N_{ar}$ ▷ Compute AR forward time

  $ar\_time = bias + k_1 \cdot G(B; \lambda RP, s) + k_2 \cdot N_{ar} + k_3 \cdot G(T_{ar}; \lambda RP, s)$

  $N_{sd} = E \cdot (1 - ((E-K)/E)^{B\gamma}), \ \ T_{sd} = B \cdot \gamma \cdot K / N_{sd}$ ▷ Compute SD forward time

  $verify\_time = bias + k_1 \cdot G(B\gamma; \lambda RP, s) + k_2 \cdot N_{sd} + k_3 \cdot G(T_{sd}; \lambda RP, s)$

  $draft\_time = draft\_bias + draft\_k \cdot G(B; \lambda RP, s)$ ▷ Compute draft model forward time

  $reject\_time = reject\_bias + reject\_k \cdot B$ ▷ Compute rejection sampling time

  $Speedup = \sigma \cdot (\gamma + 1) \cdot \frac{ar\_time}{draft\_time + ar\_time + verify\_time + \text{reject\_time}}$ ▷ Compute the speedup as formalized in Eq. 4

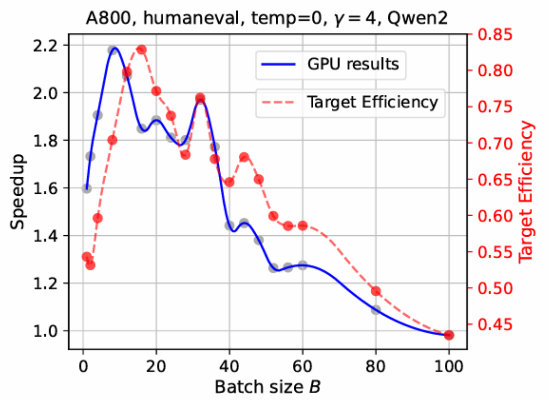
  return *Speedup*

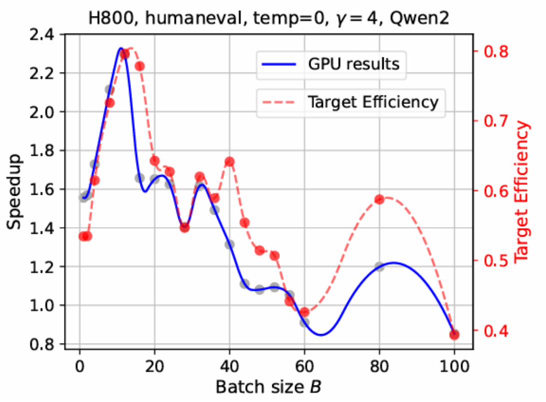
$params* = \underset{params}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{m} \left(\textbf{\textit{ComputeSpeedup}}(params, \mathbf{M}_i.B,\ \mathbf{M}_i.\gamma,\ \mathbf{M}_i.K,\ \mathbf{M}_i.E,\ \mathbf{M}_i.\sigma) - \mathbf{M}_i.Speedup \right)^2$

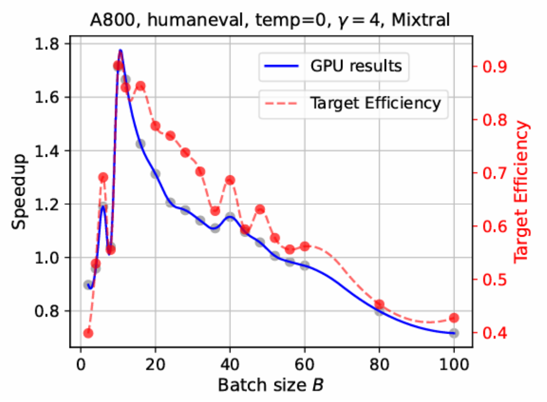
▷ Decide the optimal *params** by fitting the model to the measured inputs using the least squares criterion.

Experiment Results

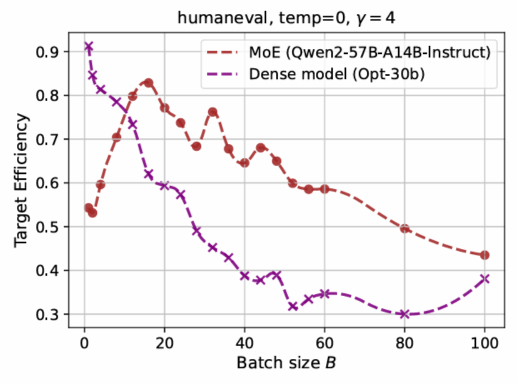
- The effectiveness of target efficiency

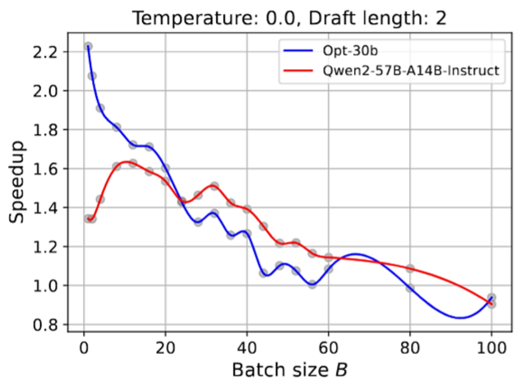


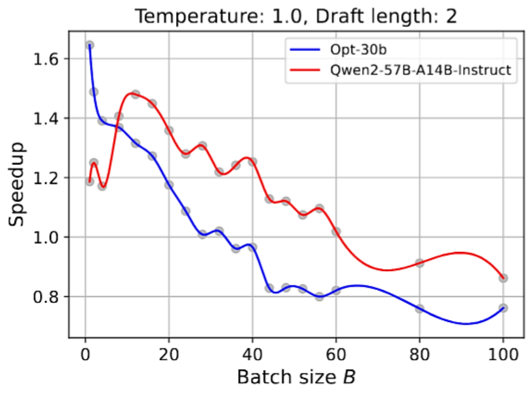




- MoE v.s. dense models

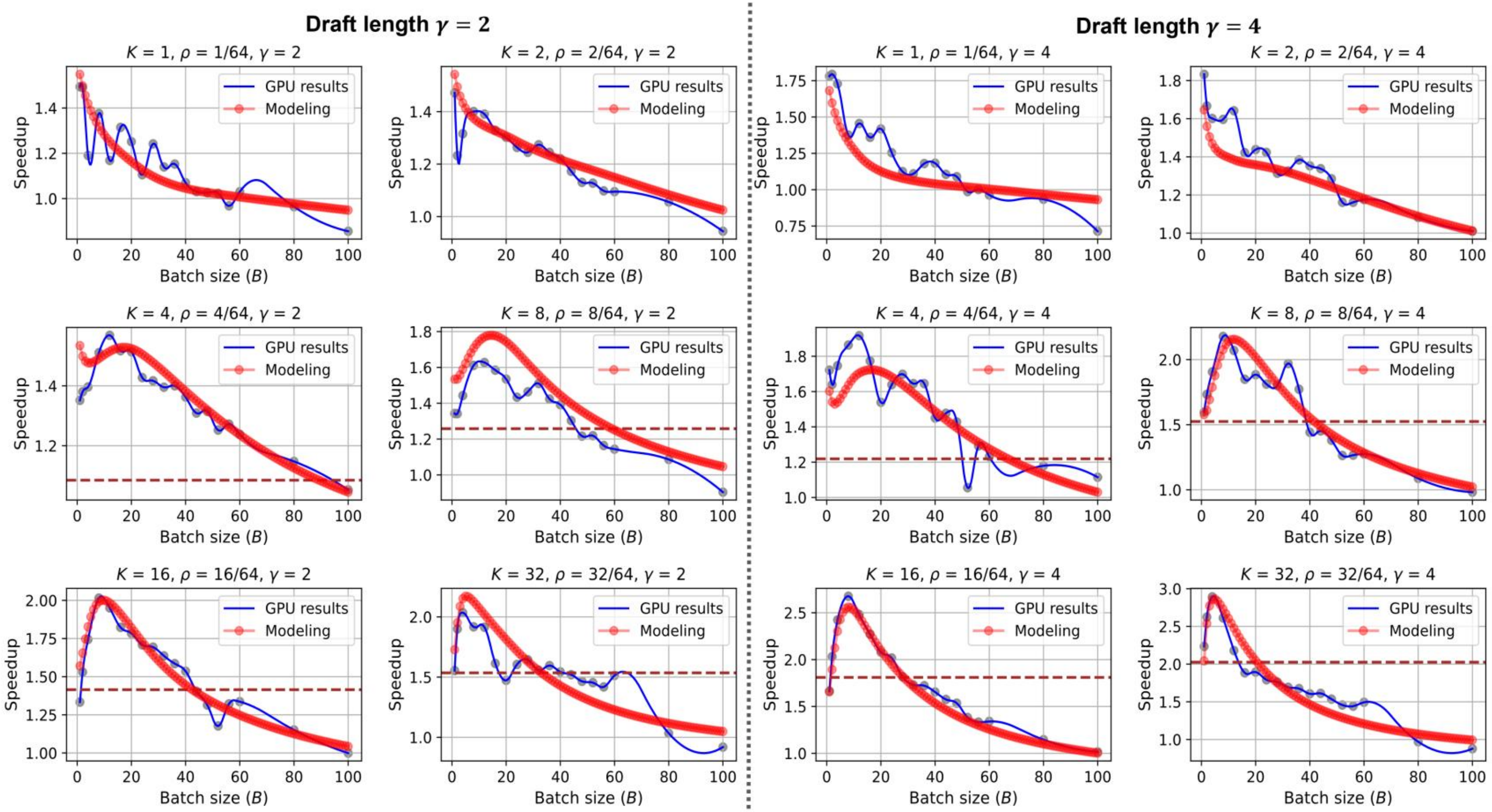






Experiment Results

- Validation of the modeling method



清華大學电子工程系
Department of Electronic Engineering, Tsinghua University

NEURAL INFORMATION PROCESSING SYSTEMS

Thank you!

Zongle Huang
huangzl23@mails.tsinghua.edu.cn
Tsinghua University, Electronic Engineering