Hardware-Aware Parallel Prompt Decoding for Memory-Efficient Acceleration of LLM Inference

Hao (Mark) Chen,
Wayne Luk, Ka Fai, Cedric Yiu,
Rui Li, Konstantin Mishchenko,
Stylianos I. Venieris, Hongxiang Fan

Speculative decoding

Once upon

Speculative decoding

Once upon a time,

Speculative decoding

Once upon a time, in a small village

Speculative decoding

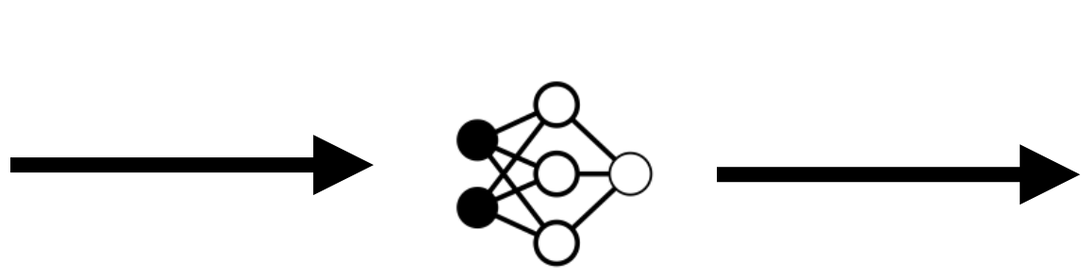
Once upon a time, in a small village

Speculative decoding

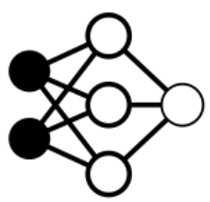
Once upon a time, in a small village

Once upon a time, there was

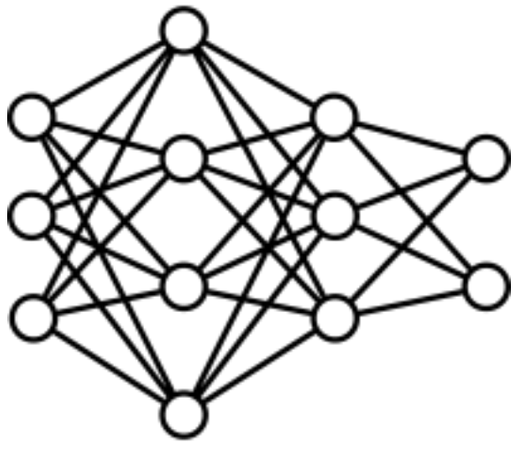
Speculative decoding

Once upon  a time, in a small village

Speculative decoding

Once upon →  → **a time, in a small village**

Once upon *a time, in a small village*

→  → **Once upon *a time*, *there***

Speculative decoding

Pros:

1. **Faster generation**
2. **Preserved quality (matching exactly the large model)**
3. **A smaller model might be already available**

Speculative decoding

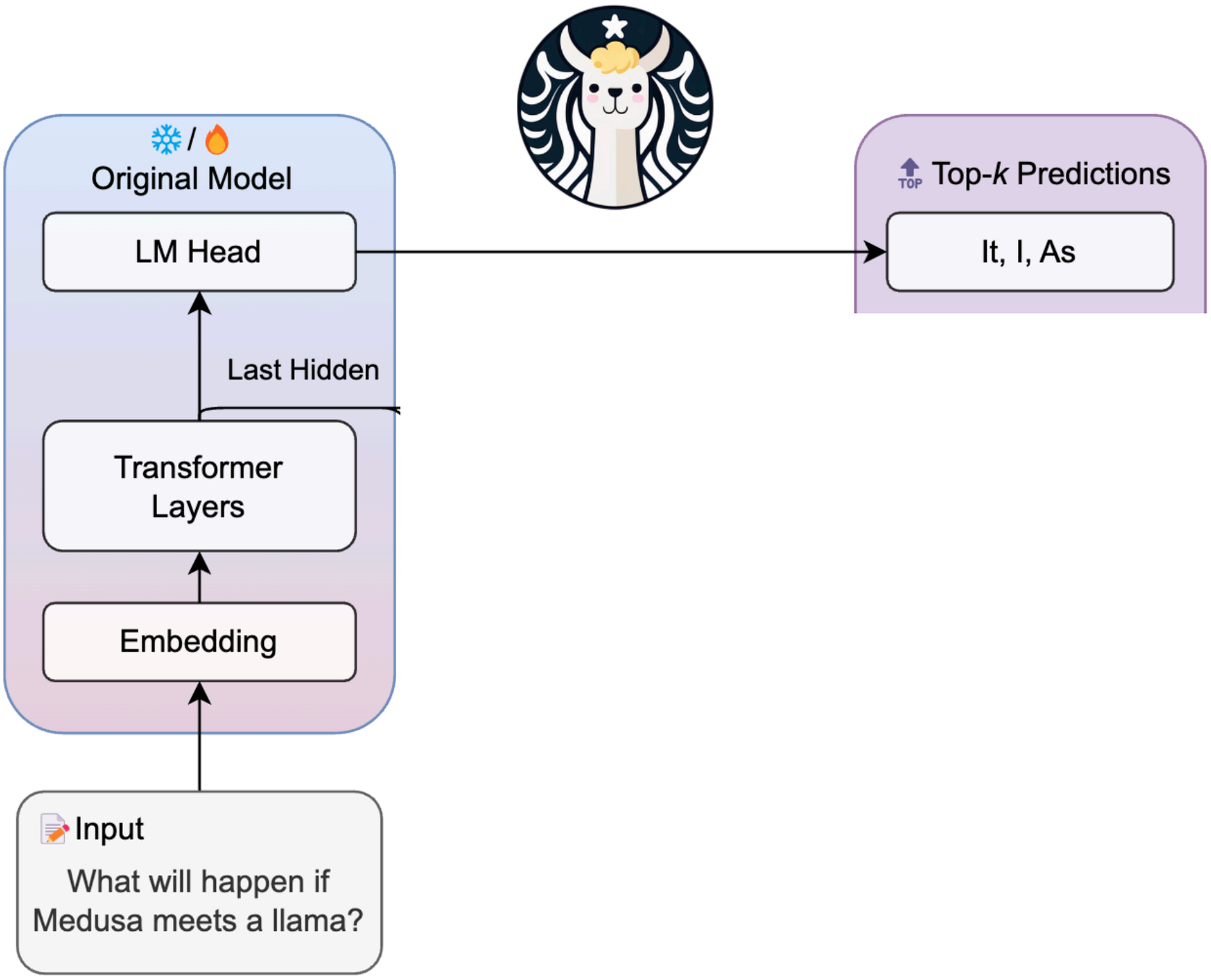
Pros:

1. **Faster generation**
2. **Preserved quality (matching exactly the large model)**
3. **A smaller model might be already available**

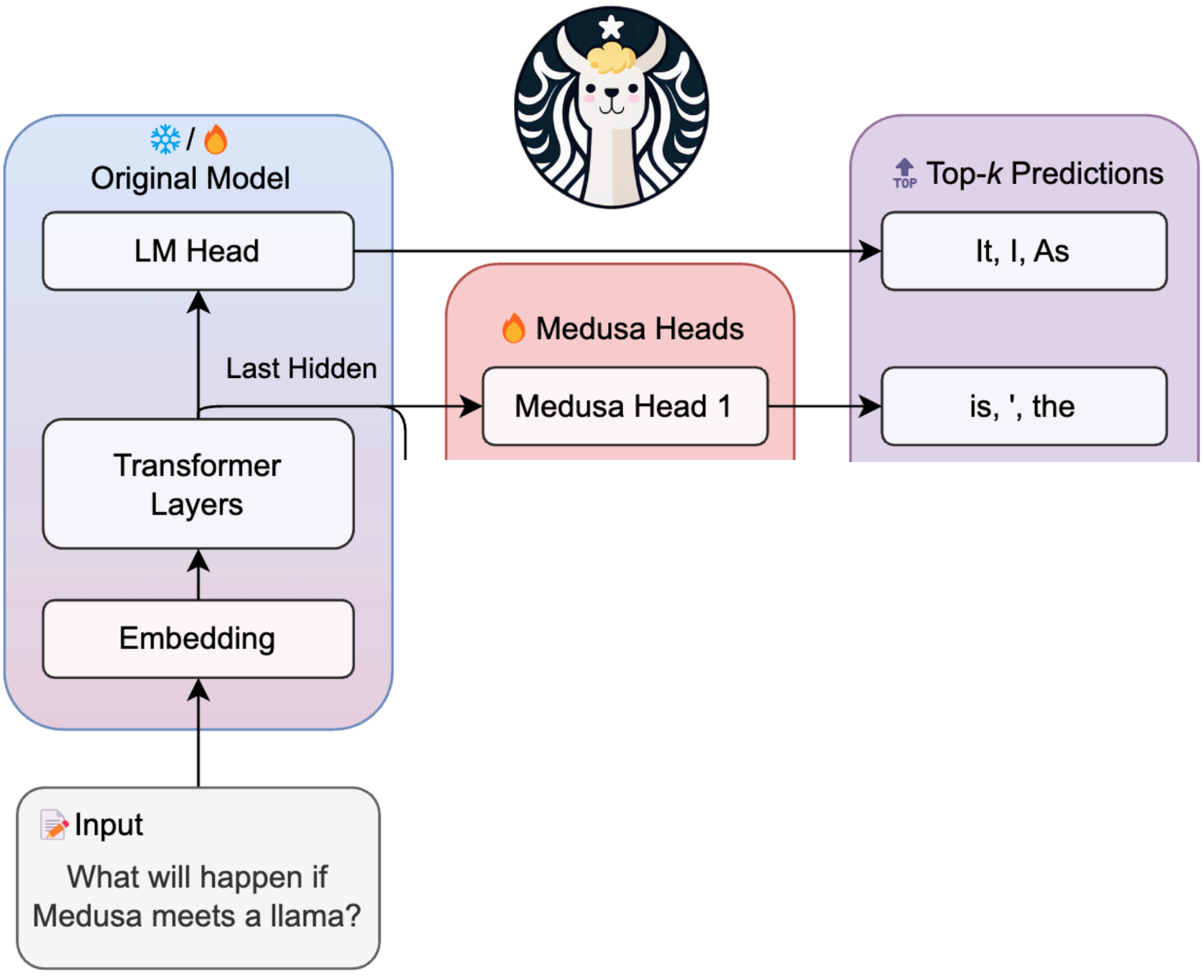
Cons:

1. **Might need to train the smaller model**
2. **Limited speed-up**
3. **More weights to store**

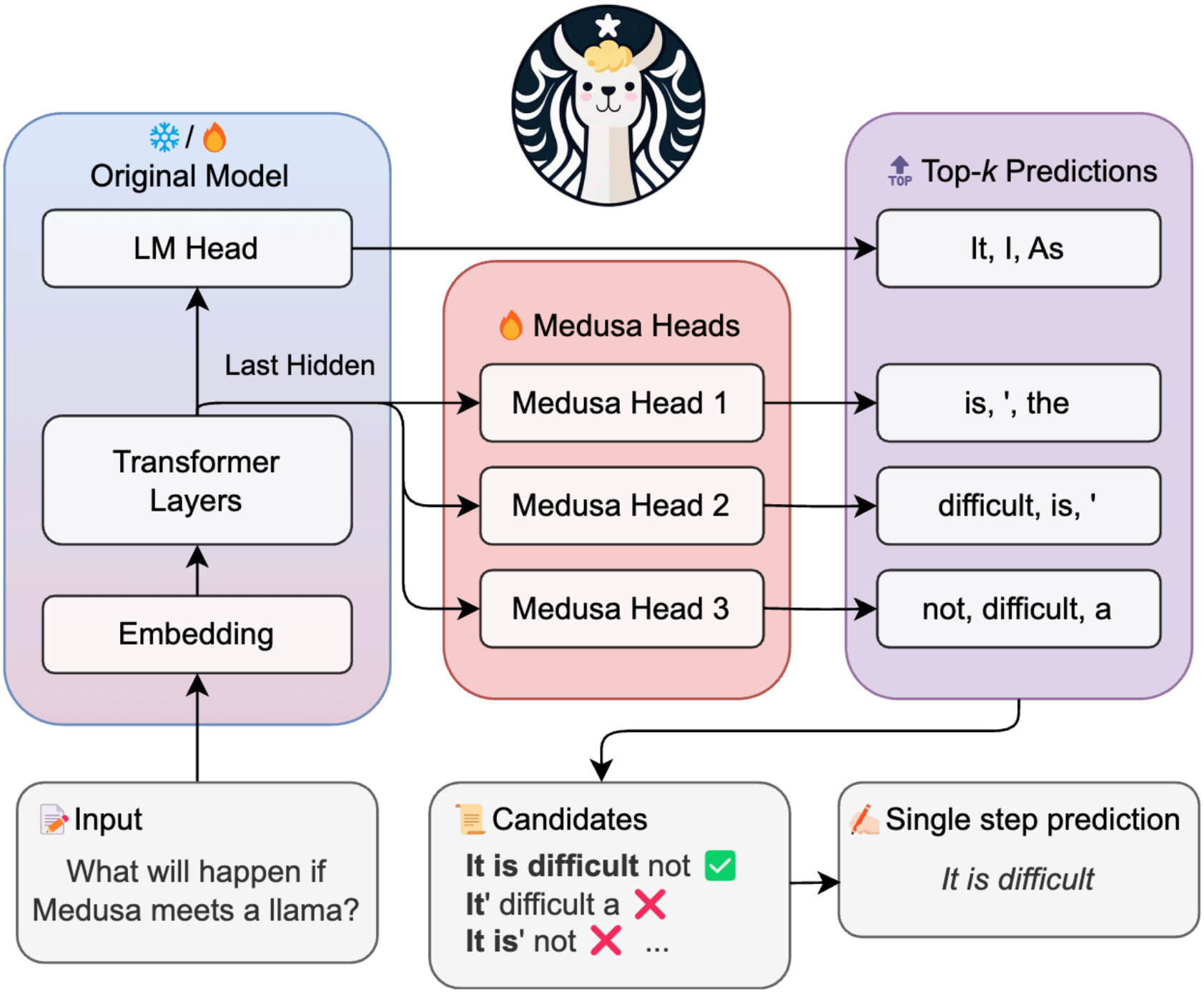
Medusa


❄ / 🔥
Original Model
LM Head
Last Hidden
Transformer Layers
Embedding
Input
What will happen if Medusa meets a llama?
Top-k Predictions
It, I, As

Medusa


❄ / 🔥
Original Model
LM Head
Last Hidden
Transformer Layers
Embedding
Input
What will happen if Medusa meets a llama?
🔥 Medusa Heads
Medusa Head 1
TOP Top-k Predictions
It, I, As
is, ', the

Medusa


❄ / 🔥
Original Model
LM Head
Last Hidden
Transformer Layers
Embedding
🔥 Medusa Heads
Medusa Head 1
Medusa Head 2
Medusa Head 3
Top-k Predictions
It, I, As
is, ', the
difficult, is, '
not, difficult, a
Input
What will happen if Medusa meets a llama?
Candidates
It is difficult not
It' difficult a
It is' not ...
Single step prediction
It is difficult

Medusa

Pros:

1. **Less training**
2. **Easy generation of multiple candidates**

Medusa

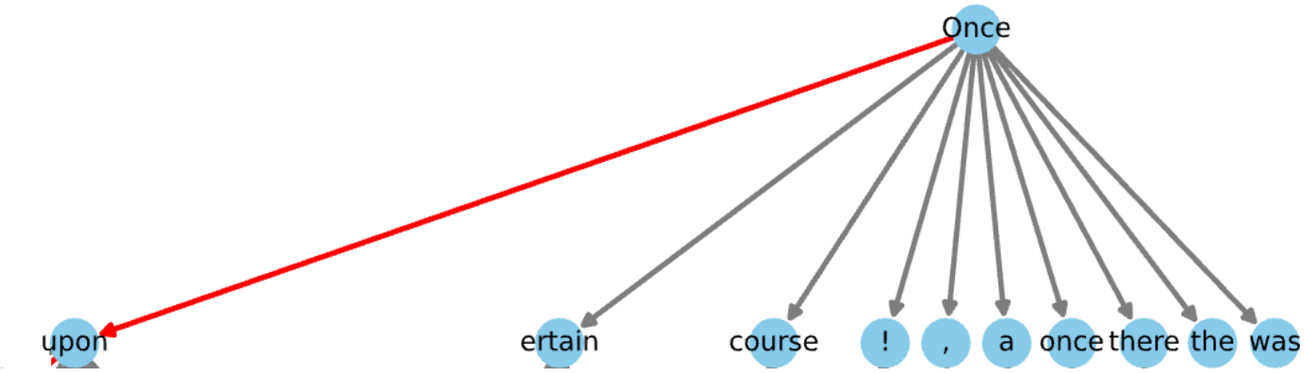
Pros:

1. Less training
2. Easy generation of multiple candidates

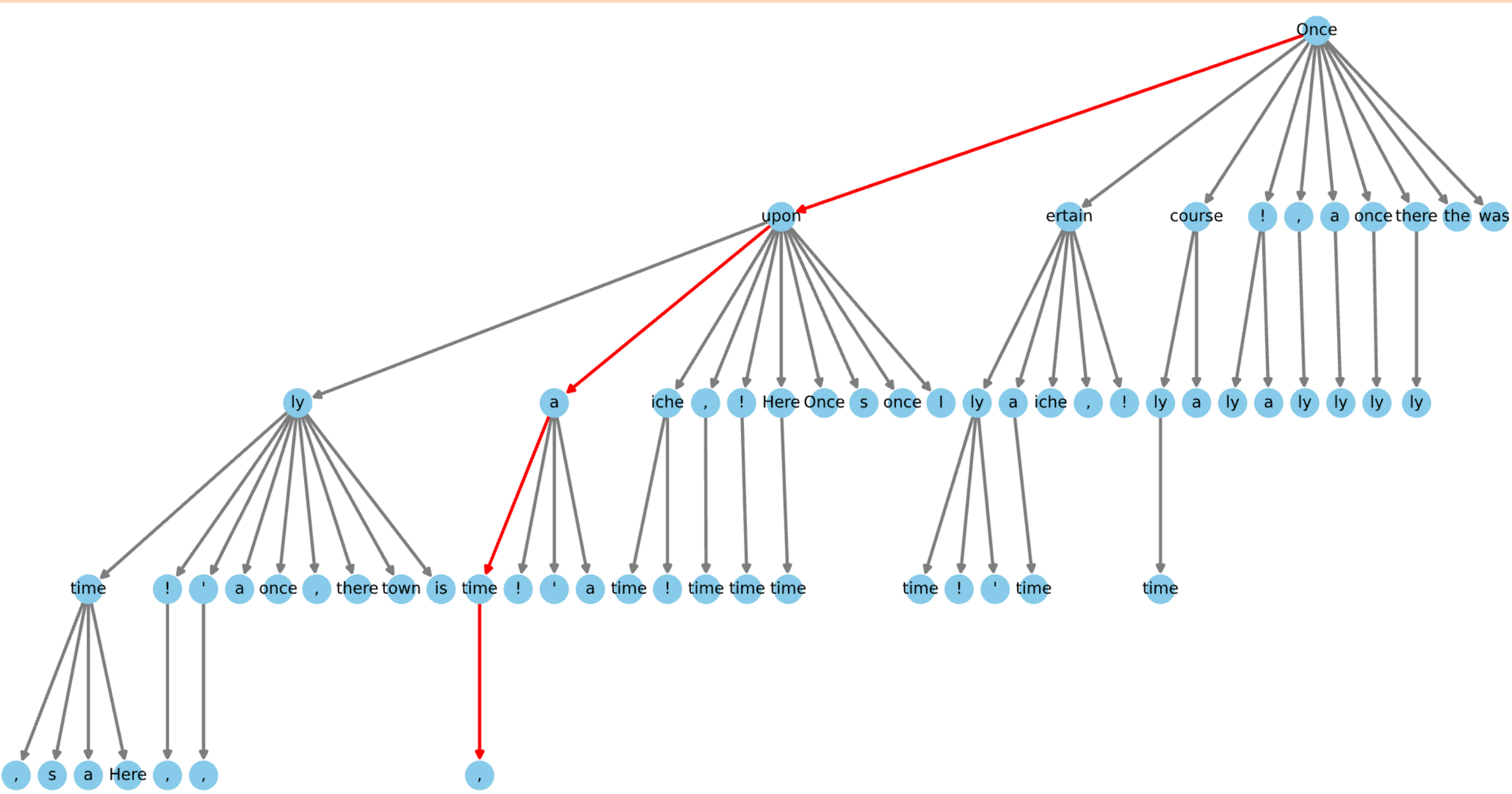
Cons:

1. Requires changing the base model for best results
2. The more tokens you generate, the worse the results
3. Medusa heads still have many parameters

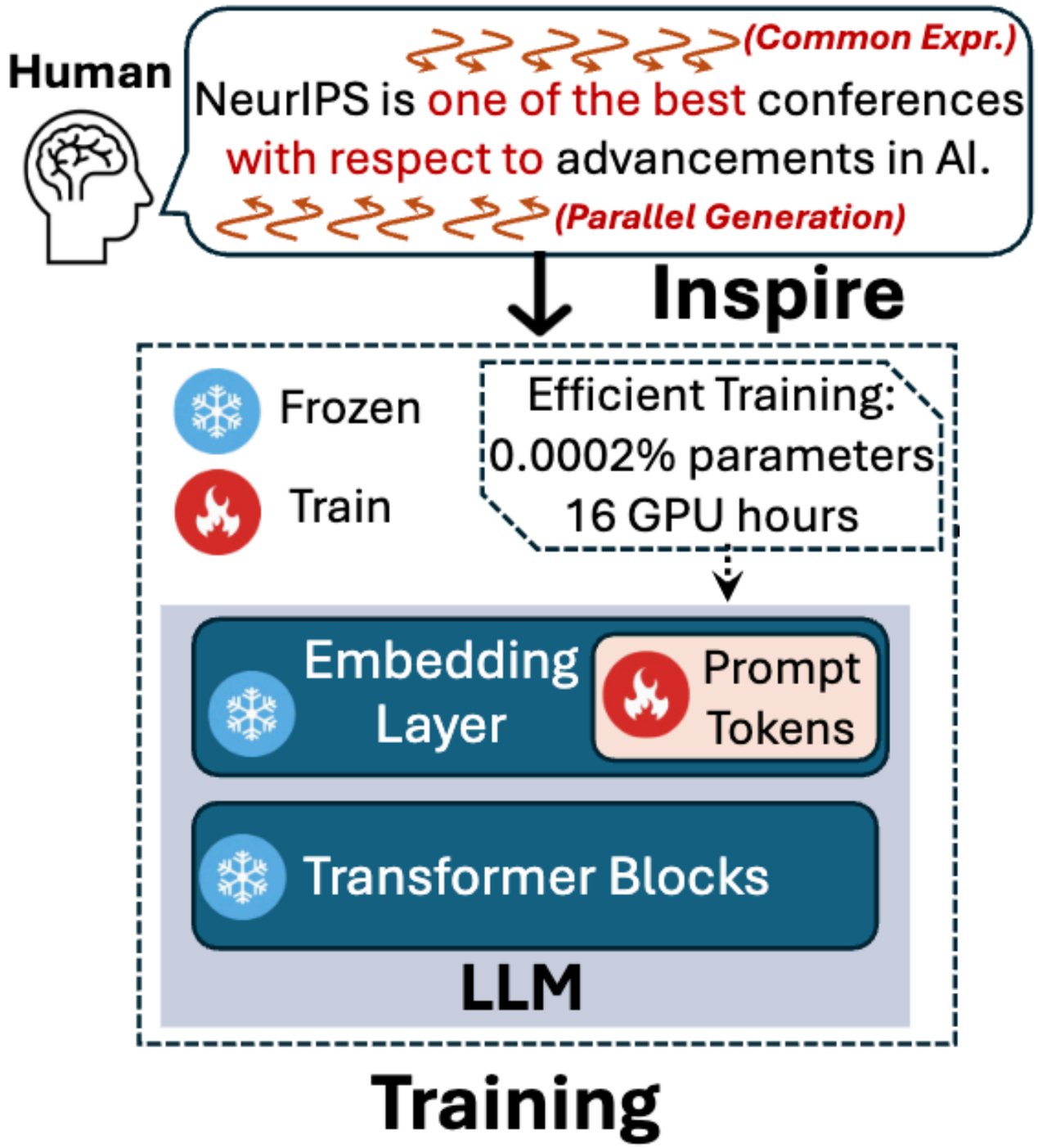
Sparse tree



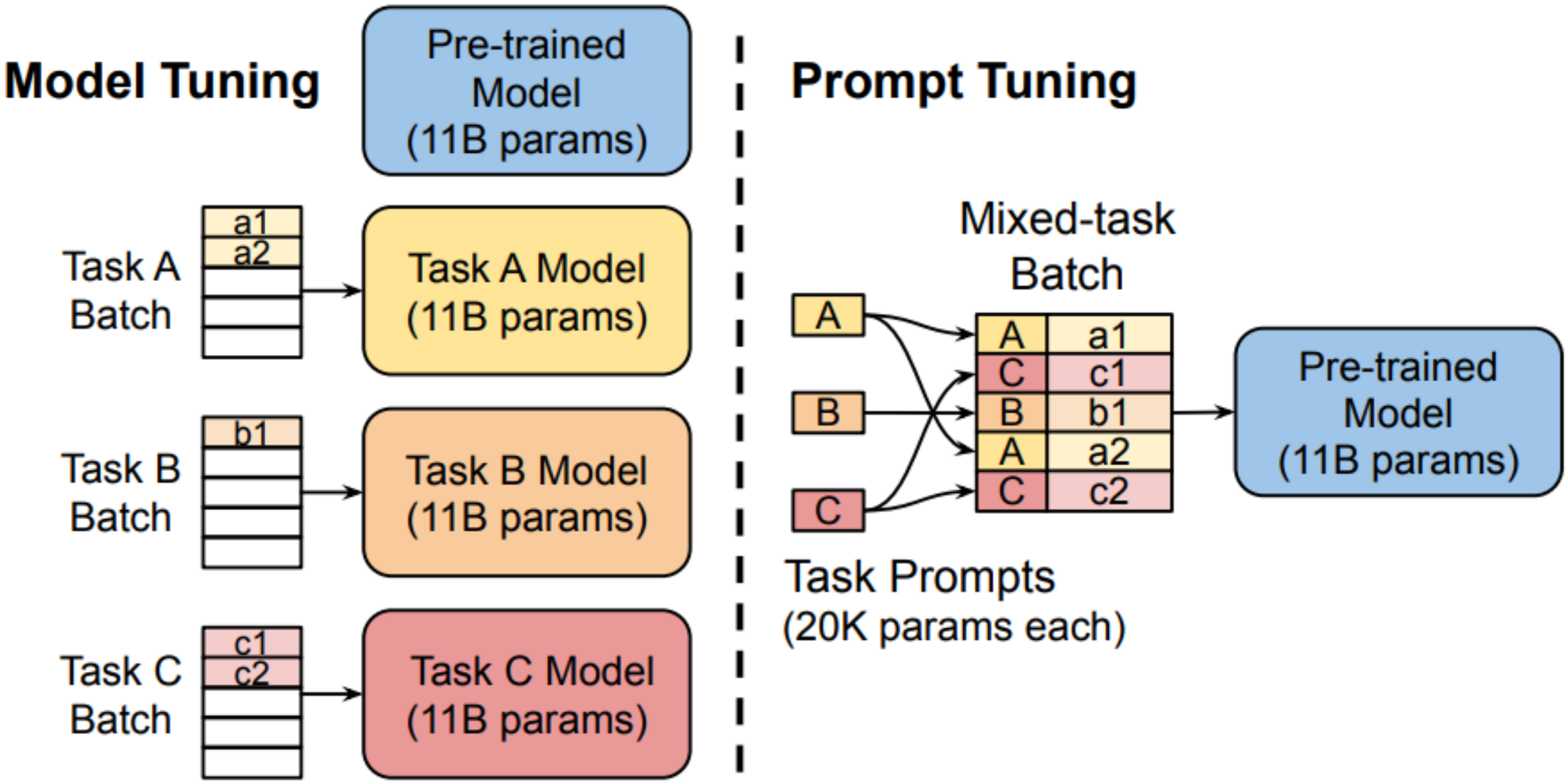
Sparse tree


Once
upon
ertain
course
!
,
a
once
there
the
was
ly
a
iche
,
!
Here
Once
s
once
I
ly
a
iche
,
!
ly
a
ly
a
ly
ly
ly
ly
time
!
'
a
once
,
there
town
is
time
!
'
a
time
!
time
time
time
time
!
'
time
time
,
s
a
Here
,
,
,

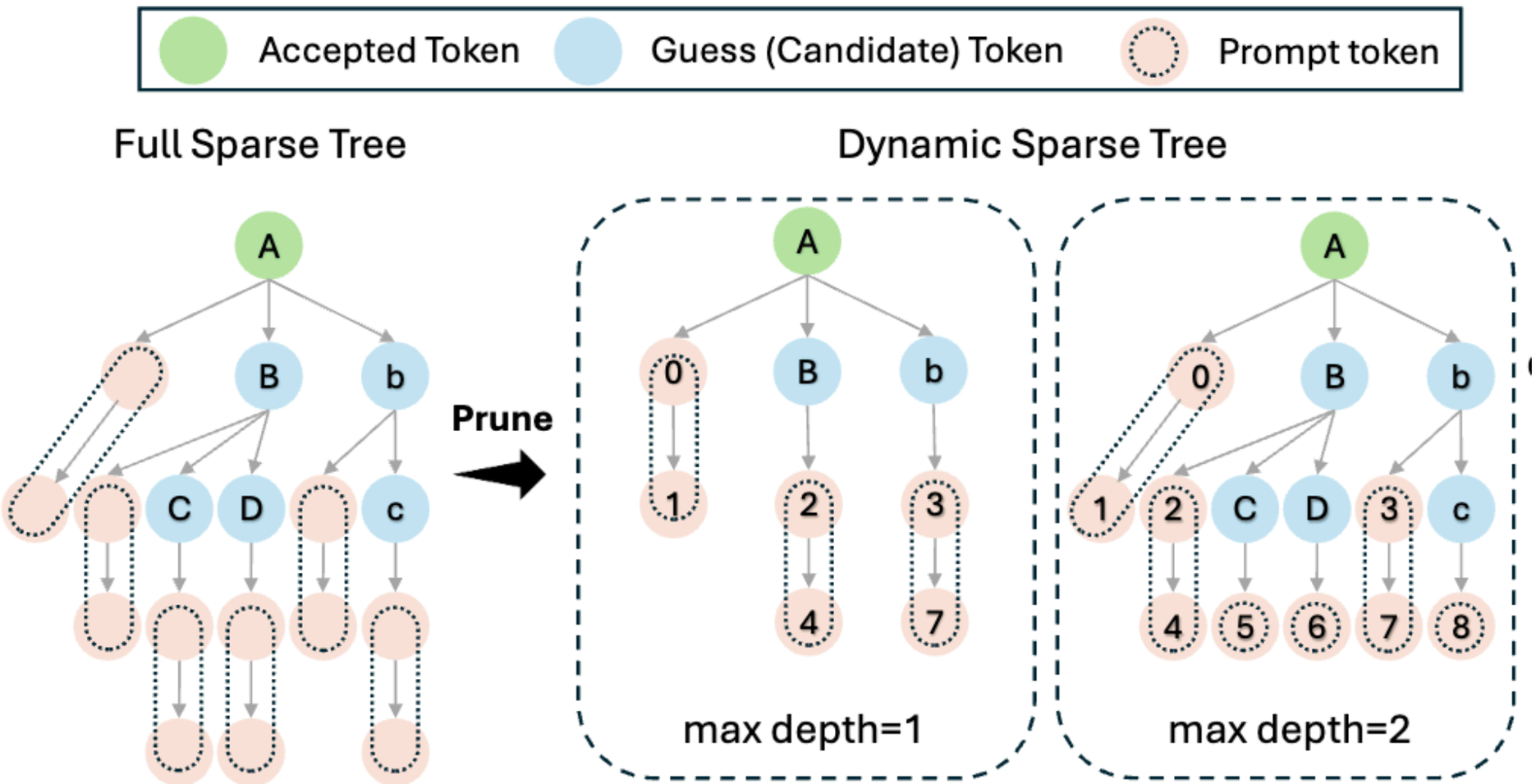
Our approach



Our approach

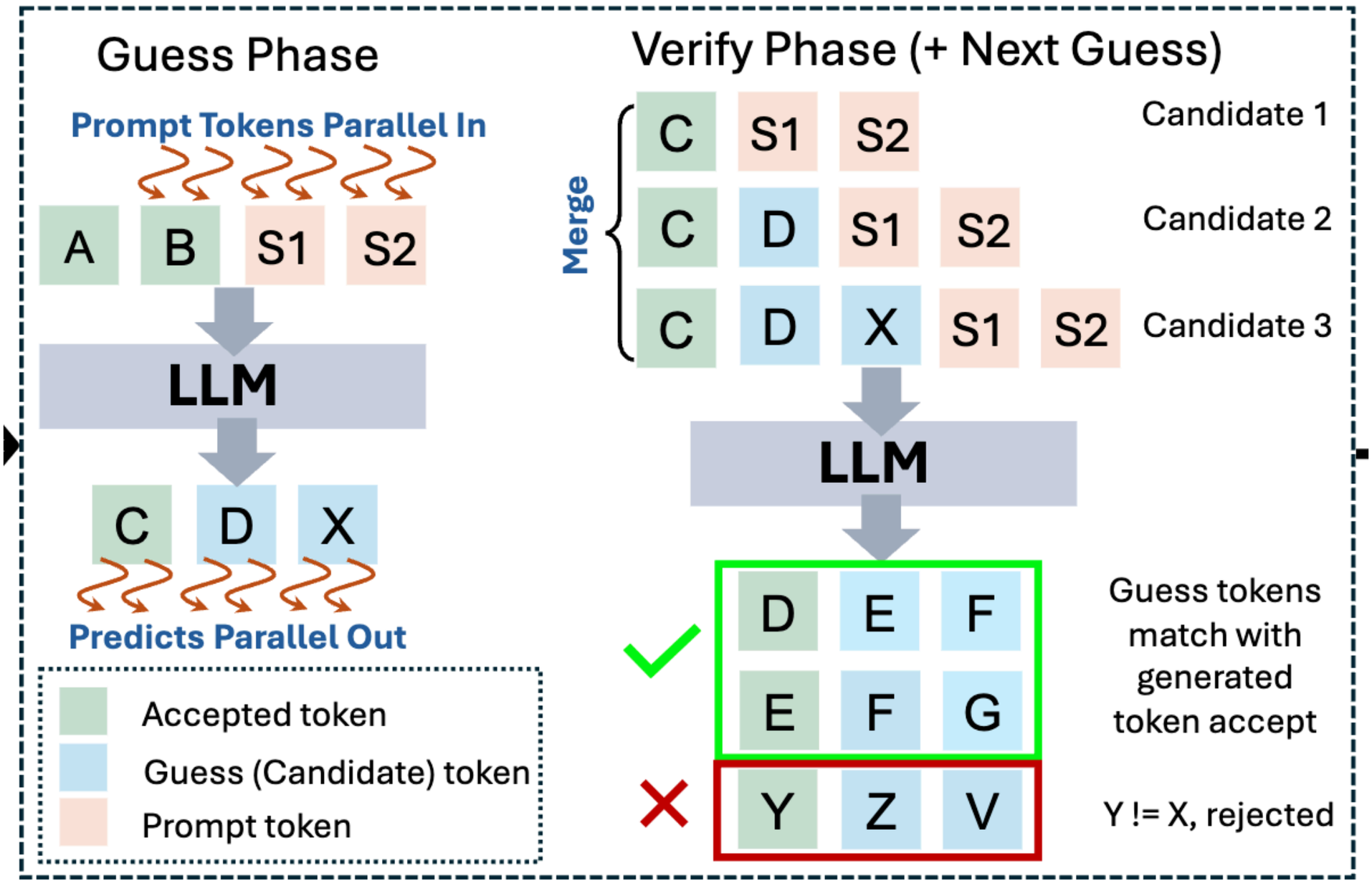


Our approach



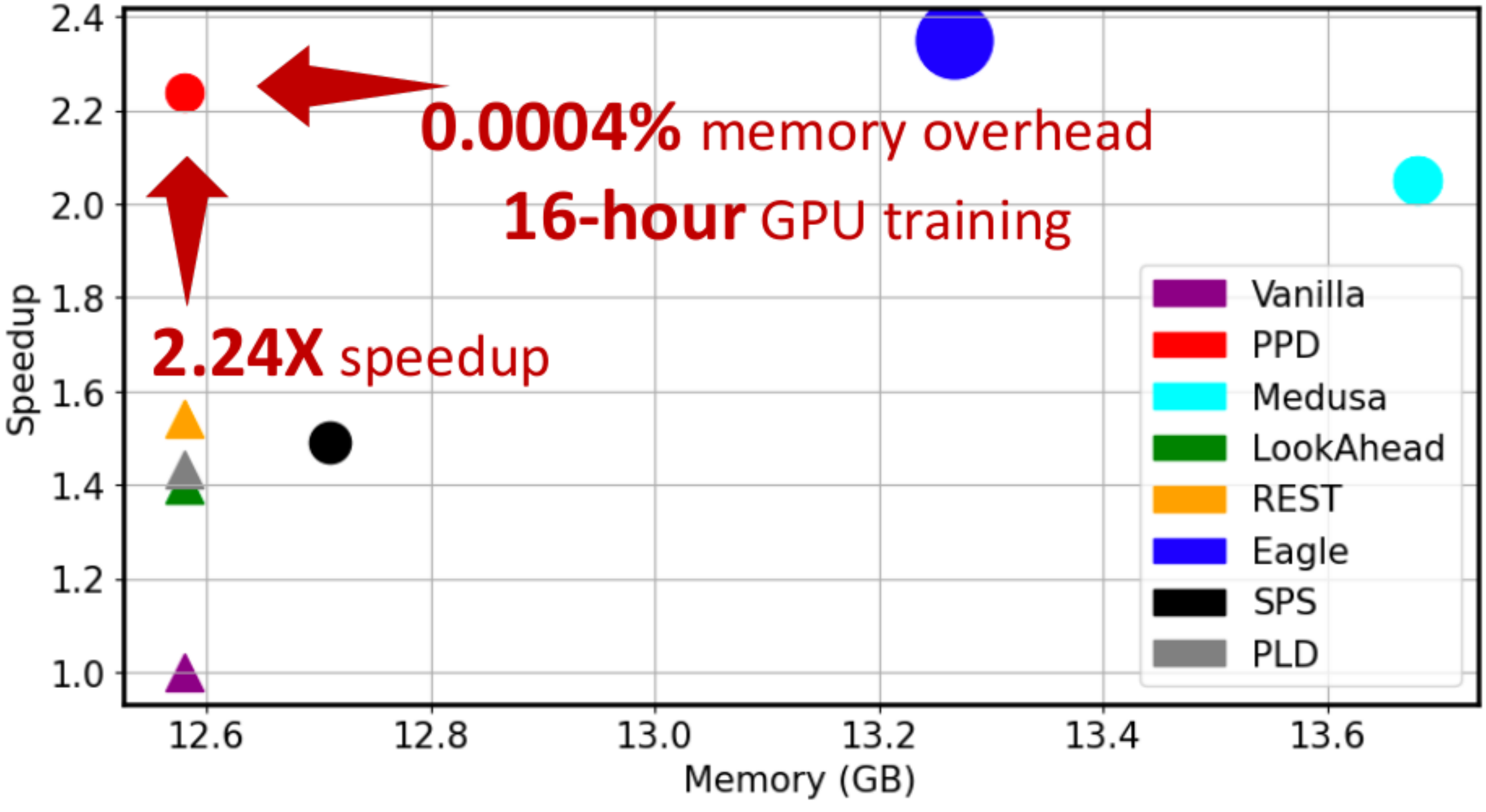
Optimize the tree using candidate probabilities.
You can additionally use speculative decoding!

Our approach



Inference Scheme

Our approach



Our approach

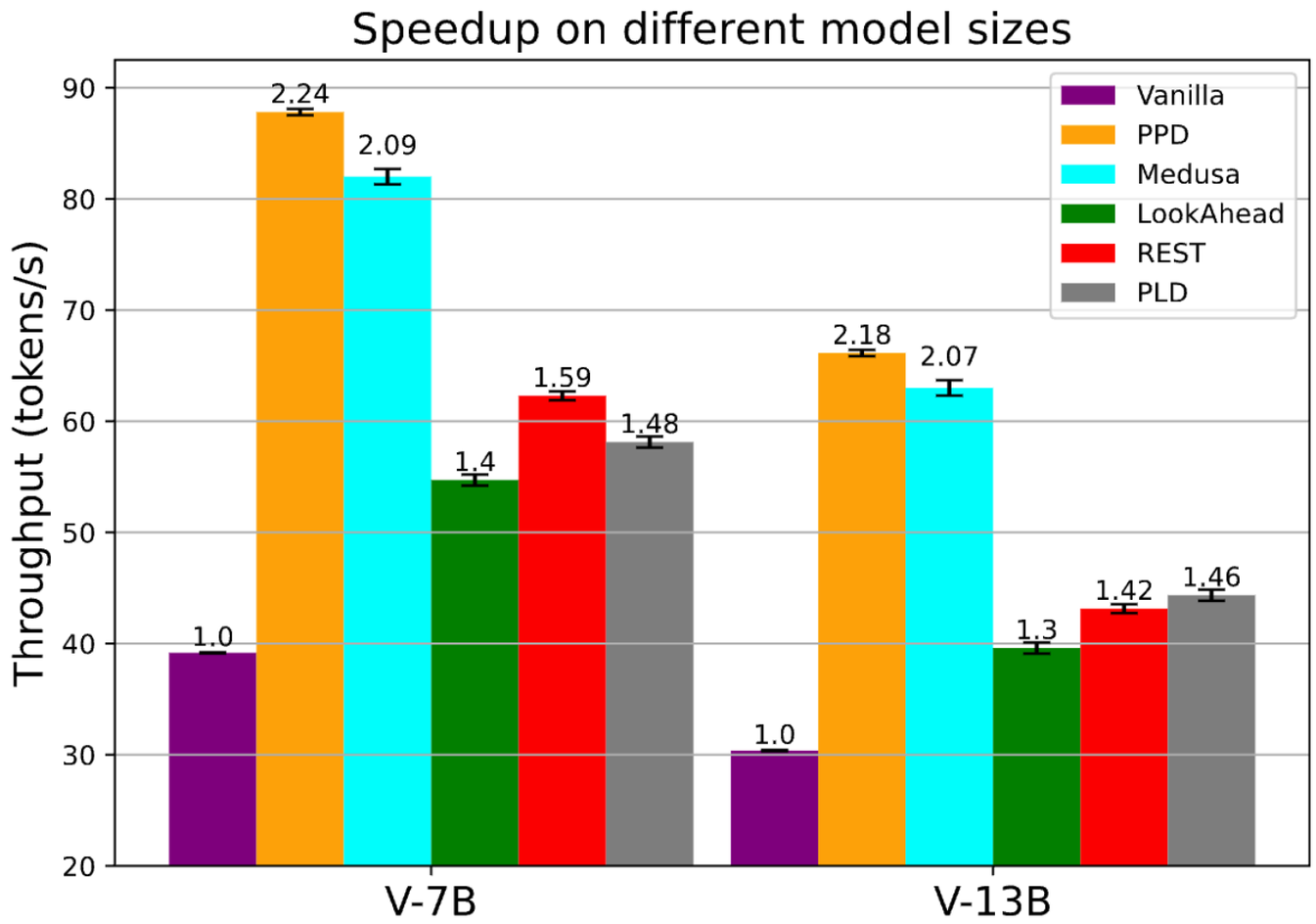


Figure 4: Comparative evaluation of latency speedup between *PPD* and other parallel decoding methods. The experiments were conducted using the MT-Bench dataset, with the temperature set to MT-Bench's default configuration for Medusa and *PPD*.

Our approach

Pros:

1. **Even less training**
2. **Less memory consumption**
3. **Higher acceptance rate**

Our approach

Pros:

1. **Even less training**
2. **Less memory consumption**
3. **Higher acceptance rate**

Cons:

1. **“Section 4 is too dense”**
2. **“Can you perform more ablations?”**
3. **“Only minor speedups over prior work”**