

## A Implementation Details

**Vanilla:** We use models from the Huggingface.transformers library with the PyTorch backend and pre-allocated KV cache. Other methods also use these models as their base.

**(Standard) Speculative Sampling:** We use the assisted generation feature from the HuggingFace Transformers library.

**PLD, Lookahead, Medusa, and Hydra:** We use the default settings and the officially released weights.

**EAGLE:** Vicuna and LLaMA2-Chat draft models use the officially released weights, while LLaMA3-Instruct is trained using the ShareGPT dataset (consistent with Medusa and Hydra).

**EAGLE-2:** For the 7B (8B), 13B, and 70B original LLMs, we set the total number of draft tokens to 60, 50, and 48, respectively, with a draft tree depth of 6, and select 10 nodes during the expansion phase.

**EAGLE-3:** EAGLE-3’s draft model achieves a significantly higher acceptance rate, allowing us to increase the draft tree depth from 6 to 8 while keeping the number of nodes the same as in EAGLE-2.

## B A Comparative Study of EAGLE-3 and HASS

The work most similar to EAGLE-3 is HASS. Both approaches simulate multi-step prediction during training, but this is neither the main focus of EAGLE-3 nor HASS. Training-time testing primarily involves adjusting the attention mask to enforce correct dependencies, which essentially simplifies tree attention into a fixed-shape form (as illustrated in Figure 6). In fact, tree attention has been widely adopted in nearly all speculative decoding methods proposed in recent years. Feeding model outputs instead of ground truth during training, known as scheduled sampling, was also widely explored in the RNN era.

The core contribution of HASS lies in identifying the train-test mismatch in EAGLE and mitigating it through tree attention-based simulation. In contrast, EAGLE-3 focuses on a different issue: the inability of EAGLE to benefit from data scaling. EAGLE-3 attributes this limitation to the feature prediction constraint—an issue also present in HASS. EAGLE-3 removes this constraint and uses tree attention for simulation. This modification enables EAGLE-3 to scale effectively with increased training data, whereas HASS does not. The ability to scale with data is the core contribution of EAGLE-3.

Figure 8 illustrates the performance of EAGLE-3 and HASS across different training data scales. Similar to EAGLE-2, HASS fails to scale up, while EAGLE-3 exhibits rapid performance improvements as more training data becomes available. Moreover, EAGLE-3 identifies that the top-layer features (used in EAGLE and subsequent works including HASS) tend to overfit to next-token prediction and are not well-suited for multi-step draft generation. To address this, EAGLE-3 replaces the top-layer features with a fusion of multi-level features. Therefore, EAGLE-3 also outperforms HASS when trained on smaller datasets.

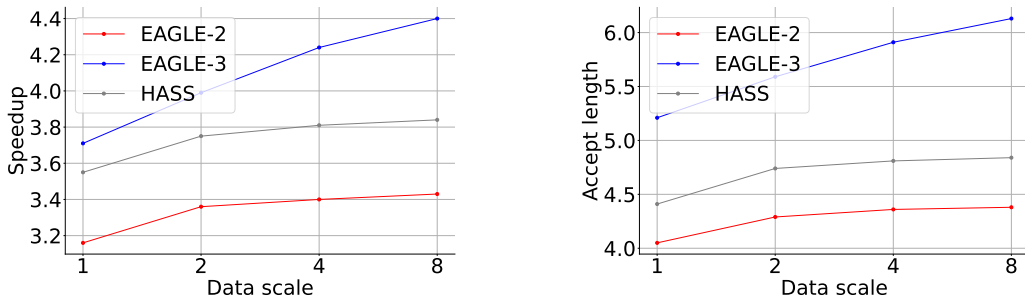


Figure 8: Scaling law evaluated on the MT-bench using LLaMA-Instruct 3.1 8B as the target model, with the x-axis representing the data scale relative to ShareGPT.