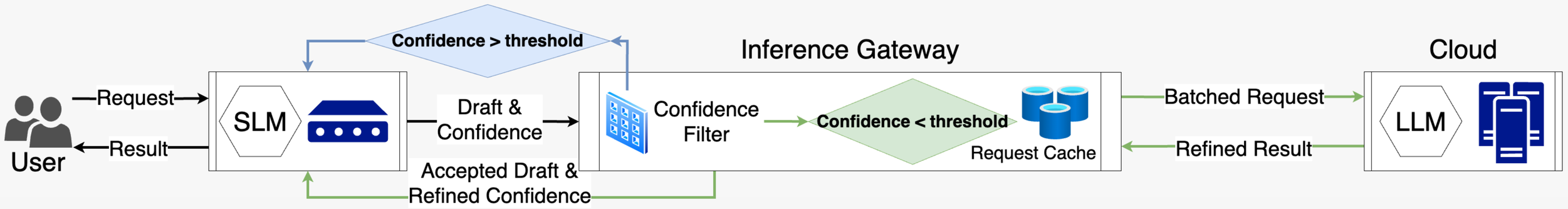
# CALID: Collaborative Accelerate LLM Inference with Draft Model with Filter Decoding

Yifan Hua[1] Shengze Wang[1] Xiaoxue Zhang[1,2] Daniel Zhu[3] Arthur Cheong[4] Karan Mohindroo[5] Allan Dewey[1] Chen Qian[1]

[1] University of California, Santa Cruz [2] University of Nevada, Reno [3] The Harker School [4]Mountain View High School [5] Pierrepont School



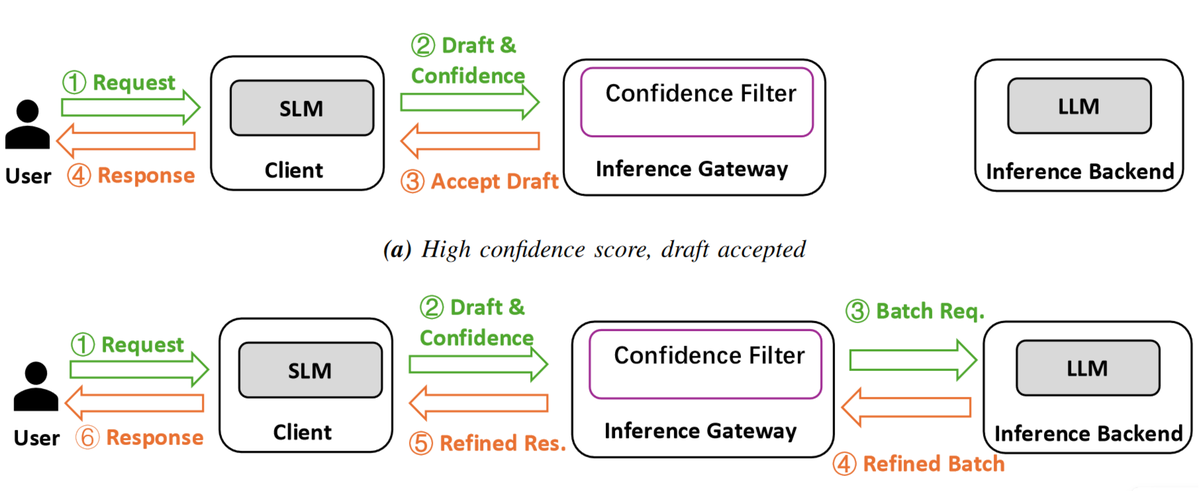

## INTRODUCTION

- The high cost of LLM processing poses barriers to widespread adoption and scalability. LLM requests can be up to 10 times more expensive than traditional keyword queries, limiting accessibility and scalability.
- Techniques like speculative decoding use smaller models for draft generation. While reducing latency in single-device settings by increasing parallelism, these methods are less effective in cloud-based environments where batching is common.

CALID Framework:

- Introduces a novel approach combining a **smaller language model (SLM)** with a confidence-based filtering mechanism.
- CALID determines when to use SLM output directly or escalate to a larger LLM based on negative **log-likelihood (NLL)** confidence scores.
- This balance optimizes resource utilization and reduces inference costs without compromising the quality of generated text.
- Offers a scalable, cost-effective solution for deploying high-quality LLM services.

## SYSTEM DESIGN




*(a) High confidence score, draft accepted*

*(b) Low confidence score, draft discarded. Large Model generates the refined result*

**Negative Log-Likelihood (NLL) based Confidence score**

Intuitively, the likelihood of the most likely candidates represents 'confidence' in the outcome from the language model itself.

$$C_1(P) = min(NLL(t_i))$$
$$C_2(P) = min(NLL(t_i)) - min_2(NLL(t_i))$$

**Filter Decoding**

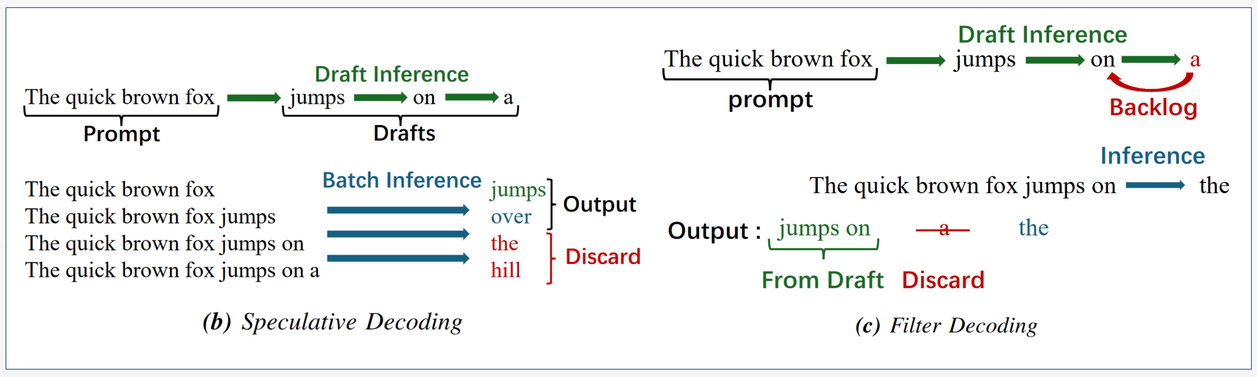
The effective confidence score can directly accept high-quality outputs generated by the SLM as the final result. In contrast, lower-quality outputs will be replaced by those of the LLM, thus optimizing the utilization of computational power. This approach ensures that computational resources are allocated efficiently, enhancing overall system performance.

**Inference Gateway**

To strike a balance between the overall generative quality and the efficiency of the Threshold is set dynamically by the inference Gateway.

$$Threshold_c = \infty, \text{ if } \eta < 1$$
$$= Percentile_\eta(\{C_i\}), \text{ if } \eta > 1$$

$\{C_i\}$ is all the historical confidence scores in the previous period L, η is the load ratio.




*(b) Speculative Decoding* *(c) Filter Decoding*

## PRELIMINARY RESULTS

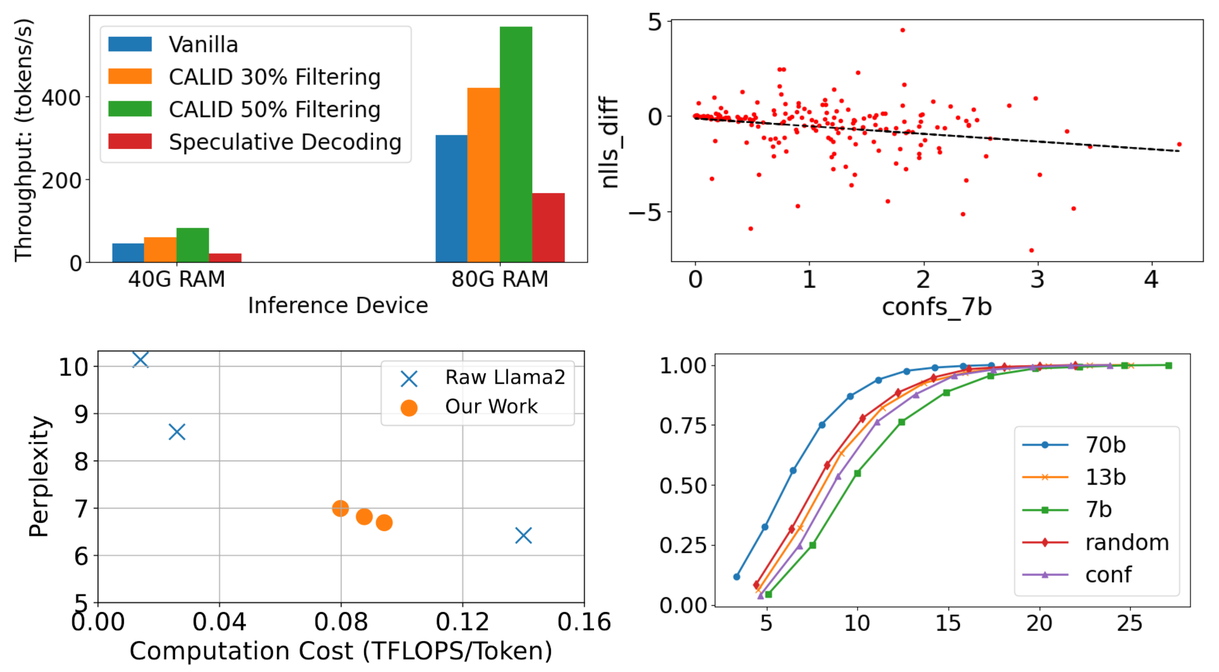
**Computation Cost vs. Perplexity**:

•**Blue crosses** represent the performance of Llama2 models (7B, 13B, and 70B from left to right).

•**Orange dots** indicate our work with various thresholds set.

•The Inference Gateway filters out 47.80%, 41.6%, and 36.4% of requests for Llama2 7B, while the remaining requests (53.2%, 58.4%, and 63.6%) are directed to Llama2 70B from left to right.

•These results demonstrate that our approach effectively reduces the computational load by filtering less confident requests, allowing for more efficient use of the Llama2 models. By optimizing request handling, we enhance overall system performance and reduce latency without sacrificing output quality.




## Conclusion

- By using the SLM to generate drafts and filtering them based on confidence scores, CALID effectively reduces the computational load on the more resource-intensive LLM. High-confidence drafts are accepted directly, while low-confidence outputs are refined by the LLM.
- Our experimental results demonstrate that CALID can significantly reduce inference costs while maintaining output quality comparable to larger models. The framework offers significant advantages for LLM service providers by optimizing computational resource allocation and reducing operational costs in cloud infrastructures.
- Promising future work includes exploring more advanced confidence scoring methods, extending CALID to support a broader range of models and applications, and investigating its integration into real-time systems to assess its impact on latency and user experience.

**References**:

[1] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.
[2] Chen, C., Borgeaud, S., Irving, G., Lespiau, J.-B., Sifre, L., & Jumper, J. (2023). Accelerating large language model decoding with speculative sampling. *arXiv:2302.01318*.
[3] Dastin, J., & Nellis, S. (2023). For tech giants, AI like Bing and Bard poses billion-dollar search problem. *Reuters*. Retrieved from Reuters
[4] Leviathan, Y., Kalman, M., & Matias, Y. (2023). Fast inference from transformers via speculative decoding. In *ICML Proceedings* (pp. 19274–19286).
[5] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., et al. (2023). Llama: Open and efficient foundation language models. *arXiv:2302.13971*.
[6] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Baskin Engineering UC SANTA CRUZ