Decoding Strategies for Diffusion-Based ASR: A Systematic Evaluation of Confidence-Based Thresholding

Jeong Hun Yeo[1] **Minsu Kim**[2,*] **Hyeongseop Rha**[1] **Yong Man Ro**[1,†]
[1] KAIST, Daejeon, Republic of Korea
[2] Google DeepMind, Tokyo, Japan
sedne246@kaist.ac.kr, ms.k@ieee.org, ryool_1832@kaist.ac.kr, ymro@kaist.ac.kr

Abstract

While LLM-based Automatic Speech Recognition (ASR) achieves high accuracy, its speed is limited by sequential autoregressive decoding. Diffusion Language Models (DLMs) offer a parallel alternative, yet their decoding strategies remain under-explored in ASR contexts. This paper analyzes three decoding schemes for DLM-based ASR: fixed-number, static confidence threshold, and dynamic confidence threshold. We introduce a round-wise analysis of decoding progress using Negative Log-Likelihood-based uncertainty as a proxy for prediction reliability. Our results show that both threshold-based strategies provide a better accuracy–speed trade-off than fixed-number schemes. This behavior is associated with the more concentrated confidence distribution observed in the evaluated ASR settings: many tokens reach high confidence early, enabling multiple tokens to be committed in early decoding rounds while lower-confidence tokens are deferred to later rounds. The static-threshold strategy achieves accuracy close to autoregressive decoding at lower decoding cost.

1 Introduction

Automatic Speech Recognition (ASR) has advanced through self-supervised pretraining (Baevski et al., 2020; Schneider et al., 2019; Hsu et al., 2021; Chen et al., 2022), large-scale weakly supervised training (Radford et al., 2023; Pratap et al., 2024; Peng et al., 2024), and LLM-based decoder modeling (Grattafiori et al., 2024; Bai et al., 2023; Chen et al., 2024; Wu et al., 2023; Chu et al., 2023). However, inference in LLM-based ASR remains computationally expensive, primarily because autoregressive (AR) decoding generates tokens sequentially, requiring repeated forward passes through large decoder models.

Before the adoption of LLMs, non-autoregressive ASR methods such as Imputer (Chan et al., 2020), Mask-CTC (Higuchi et al., 2020), and Align-Refine (Chi et al., 2021) had already reduced reliance on sequential decoding through iterative parallel prediction and refinement of token sequences and latent alignments. This parallel-refinement paradigm has since been revisited in ASR (Wang et al., 2026; Tian et al., 2026) through the adoption of Diffusion Language Models (DLMs) (Nie et al., 2025; Ye et al., 2025). These models leverage linguistic knowledge acquired through large-scale language pretraining and enable bidirectional parallel decoding by iteratively predicting multiple masked positions and committing high-confidence tokens.

Since token commitment in DLMs relies on heuristic confidence criteria, prior work has explored fixed-number, static-threshold, and dynamic-threshold decoding strategies (Fu et al., 2026; Ben-Hamu et al., 2025; Nie et al., 2025; Chang et al., 2022; Yu et al., 2025; Wu et al., 2026). Existing DLM-based ASR systems, however, employ individual strategies in isolation: Whisper-LLaDA (Wang et al., 2026) uses fixed-number decoding, whereas dLLM-ASR (Tian et al., 2026) uses static thresholding. Neither systematically compares alternative commitment strategies nor analyzes why their behavior differs. We therefore compare the three strategies in terms of accuracy, decoding speed, and round-wise progress.

Concretely, we analyze decoding progress through prediction reliability and token throughput. Since insertion and deletion errors break alignment across decoding rounds, round-wise accuracy cannot be measured directly. We therefore use cumulative token-level Negative Log-Likelihood (NLL) as a proxy for prediction reliability, with an AR configuration of the same DLM as a reference. We measure throughput by counting newly unmasked

*Work done in an advisory role only. †Corresponding author.

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 41652–41663
October 24-29, 2026 ©2026 Association for Computational Linguistics

tokens per round and examine confidence distributions to characterize differences in commitment behavior.

Our main findings are threefold. First, static thresholding offers a favorable overall accuracy–speed trade-off across test sets and two DLM backbones, achieving comparable WER to dynamic thresholding with lower RTF, while both threshold-based strategies outperform fixed-number decoding in the evaluated settings. Second, round-wise analyses of NLL and token throughput indicate that the differences underlying the final efficiency and accuracy gaps emerge primarily in the early decoding rounds, where threshold-based strategies commit many high-confidence tokens. Third, this behavior is associated with confidence scores being more concentrated at high values in the evaluated ASR settings than in the examined text tasks, including math reasoning, machine translation, and summarization, which is consistent with the observed early commitment behavior.

2 Baseline DLM-based ASR Model

2.1 DLM-based ASR

Inspired by Whisper-LLaDA (Wang et al., 2026), we build a DLM-based ASR system that combines Whisper-medium.en (Radford et al., 2023) as the speech encoder with LLaDA-8B-Instruct (Nie et al., 2025) as the decoder. A lightweight temporal adapter downsamples and projects the speech representations into the DLM embedding space. During training, the Whisper encoder is frozen, while the DLM is fine-tuned via LoRA (Hu et al., 2022) and the adapter is trained jointly.

Following the diffusion formulation (Nie et al., 2025), we randomly mask the target sequence x_0 with a ratio $t \sim \mathcal{U}(0, 1)$ to obtain x_t, and train the model to reconstruct masked tokens conditioned on audio features a and unmasked text context. The training objective is:

$$\mathcal{L} = -\mathbb{E}_{t,a,x_0,x_t}\left[\frac{1}{t}\sum_{i=1}^{L} m_i \log p_\theta(x_0^i \mid a, x_t)\right], \tag{1}$$

where $m_i = \mathbf{1}[x_t^i = \mathrm{M}]$, L is the sequence length, M is the mask token, and θ denotes trainable parameters. We train on LibriSpeech 960h (Panayotov et al., 2015) for 100,000 steps (≈4.5 epochs) using 8 NVIDIA RTX A6000 GPUs, and evaluate on test-clean.

2.2 Block-based Decoding

The decoder operates on a fixed output canvas of $L = 128$ mask tokens. We use block-based decoding, where the canvas is divided into blocks of size B. The blocks are processed sequentially, while diffusion unmasking is performed within each block. Fully parallel decoding corresponds to $B = L$. As in Whisper-LLaDA (Wang et al., 2026), once an end-of-sequence (EOS) token is committed, all remaining positions are set to EOS and subsequent blocks are skipped.

3 Analysis Method

3.1 Confidence Estimation in Parallel Decoding

In DLM-based ASR, decoding iteratively refines masked tokens. At each round r, for each masked position i where $x_r^i = \mathrm{M}$, the model predicts a token conditioned on the audio features a and the current context x_r:

$$\hat{x}_{r+1}^i = \arg\max_{v \in \mathcal{V}} p_\theta(v \mid a, x_r), \tag{2}$$

where $\mathcal{V}$ is the vocabulary. We define the confidence of this prediction as $c^{(i)} = p_\theta(\hat{x}_{r+1}^i \mid a, x_r)$.

3.2 Target Decoding Strategies

Following prior analyses of confidence-based parallel decoding (Fu et al., 2026), we evaluate three commonly used commitment schemes. Based on confidence scores $c^{(i)}$, we select a subset of masked positions to unmask and fix for subsequent rounds.

Fixed-number (top-k) scheme. The model unmasks the k most confident tokens among the currently masked positions in each round (Nie et al., 2025).

Static confidence threshold. The model unmasks all tokens whose confidence exceeds a threshold $C \in (0, 1)$ (Yu et al., 2025). If none exceed C, the most confident token is unmasked to ensure progress.

Dynamic confidence threshold. Following Wu et al. (2026), confidence scores over the current m masked positions are sorted as $c^{(1)} \geq \cdots \geq c^{(m)}$, and the largest k satisfying

$$(k + 1)(1 - c^{(k)}) < f \tag{3}$$

is selected, where $f > 0$ is a predefined factor. The top-k tokens are then unmasked; if no k satisfies the criterion, the most confident token is unmasked.

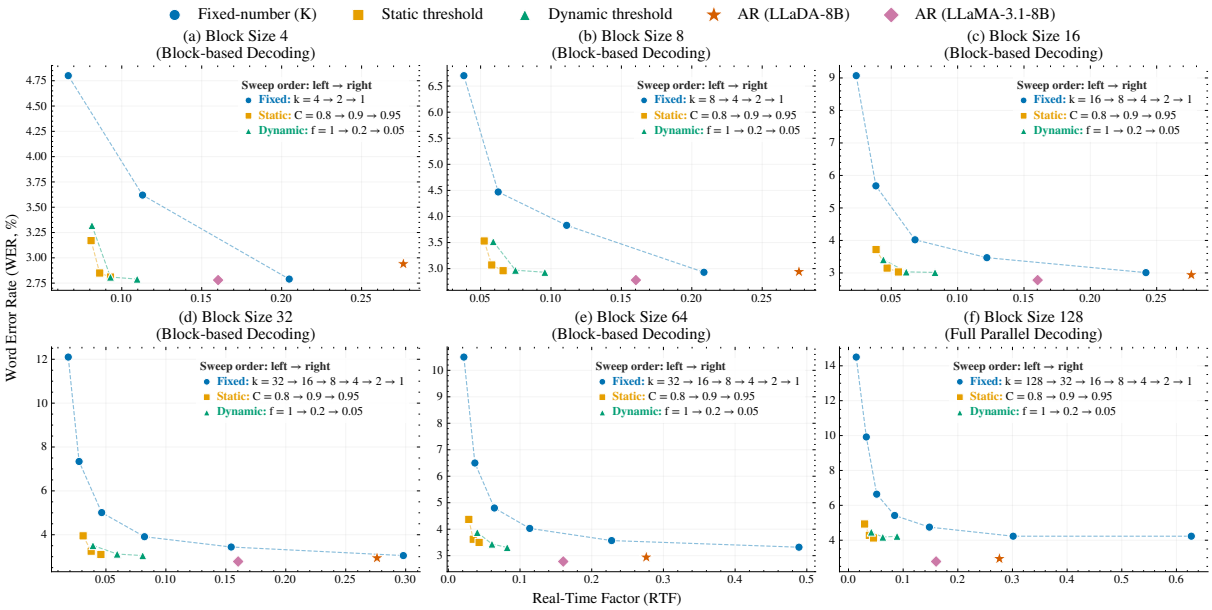


Figure 1: WER–RTF trade-off across decoding strategies and block sizes in DLM-based ASR. Lower WER and RTF are preferred, so points closer to the origin indicate better trade-offs.

3.3 Quantifying Progress via Token-level Uncertainty

To analyze decoding progress, we use token-level Negative Log-Likelihood (NLL) as a proxy for uncertainty. For each predicted token $\hat{x}^i_{r+1}$ at round r, we define

$$U^i_r = -\log p_\theta(\hat{x}^i_{r+1} \mid a, x_r). \tag{4}$$

Let $\mathcal{C}_r$ be the set of tokens committed up to round r. The cumulative uncertainty is

$$U^{cum}_r = \sum_{i \in \mathcal{C}_r} U^i_{round(i)}, \tag{5}$$

where $round(i)$ denotes the round in which token i was fixed. We plot U^{cum}_r against normalized progress $P_r = |\mathcal{C}_r|/L$.

As a reliability reference, we use an AR configuration of the same DLM, which unmasks tokens sequentially from left to right:

$$U^{cum}_{AR}(P) = \sum_{i=1}^{\lfloor PL \rfloor} -\log p_\theta(\hat{x}^i_{AR} \mid a, \hat{x}_{<i}), \tag{6}$$

where $\hat{x}^i_{AR}$ is the predicted token at position i and $\hat{x}_{<i}$ is the generated prefix. Deviations from U^{cum}_{AR} serve as an indicator of the additional uncertainty associated with incomplete context in parallel decoding.

4 Experiments

For the main analysis metrics, we measure WER and inference speed using the Real-Time Factor (RTF), defined as the total inference time divided by the input audio duration. All latency measurements are performed on a single idle NVIDIA RTX A6000 with batch size 1, using the dual KV-cache of Fast-dLLM (Wu et al., 2026). As AR reference points, we use two baselines: AR (LLaDA-8B), which decodes the same DLM one token at a time from left to right, and AR (LLaMA-3.1-8B) (Grattafiori et al., 2024), a LoRA-tuned autoregressive decoder using the same speech encoder and adapter. Additional numerical results are provided in the Appendix.

4.1 Decoding Strategies and Accuracy-Speed Trade-offs

We first evaluate the three diffusion-based decoding strategies on LibriSpeech test-clean by sweeping their hyperparameters: fixed-number decoding ($k \in \{1, 2, 4, 8, 16, 32, 128\}$), static thresholding ($C \in \{0.8, 0.9, 0.95\}$), and dynamic thresholding ($f \in \{1.0, 0.2, 0.05\}$), together with two AR baselines (Fig. 1). For block sizes up to 64, stricter commitment criteria (*i.e.*, smaller k, higher C, or lower f) consistently reduce WER at the cost of higher RTF, reflecting the accuracy–speed trade-off in diffusion-based decoding. Compared with fixed-number decoding, threshold-based strategies achieve comparable WER with lower RTF over much of the evaluated range. Static and dynamic thresholding show similar recognition accuracy, while static thresholding often provides lower RTF, especially in Fig. 1(b,c).

Test set	Static (C=0.95)	Dynamic (f=0.2)	Fixed-k (k=2)
In-domain			
test-other	6.34 (.072)	6.34 (.079)	6.88 (.134)
Out-of-domain			
TED-LIUM 3	10.96 (.059)	10.97 (.066)	11.21 (.131)
AMI-IHM	25.78 (.167)	25.79 (.172)	26.04 (.226)

Table 1: Cross-dataset comparison of commitment strategies. Each cell reports WER (%) with RTF in parentheses.

Test set	Backbone	Static (C=0.95)	Dynamic (f=0.2)	Fixed-k (k=4)
test-clean	LLaDA-8B	3.11 (.045)	3.12 (.054)	3.91 (.080)
	Dream-7B	3.62 (.050)	3.59 (.055)	4.36 (.076)
test-other	LLaDA-8B	6.45 (.066)	6.43 (.075)	7.60 (.087)
	Dream-7B	6.81 (.070)	6.80 (.075)	7.79 (.081)

Table 2: Decoder backbone comparison on LibriSpeech. Each cell reports WER (%) with RTF in parentheses.

Static thresholding achieves accuracy close to the AR (LLaMA-3.1-8B) baseline at lower RTF. With block size 4 and $C = 0.95$, it achieves 2.81% WER compared with the baseline WER of 2.78%, with a $1.6\times$ speedup in terms of RTF. As the block size increases, decoding generally becomes faster and less accurate; in the fully parallel setting, static thresholding achieves 4.13% WER with a $3.5\times$ speedup over the same AR baseline.

4.2 Cross-Dataset Generalization of Decoding Strategies

To examine whether the accuracy–speed trade-off observed on LibriSpeech test-clean extends to more challenging conditions, we further evaluate the three decoding strategies with a block size of 16 on LibriSpeech test-other (Panayotov et al., 2015), TED-LIUM 3 (Hernandez et al., 2018), and AMI-IHM (Carletta et al., 2006), covering more challenging in-domain read speech, out-of-domain prepared speech, and out-of-domain spontaneous meeting speech, respectively.

As shown in Table 1, threshold-based strategies yield lower WER and RTF than fixed-number decoding across all three test sets. At nearly matched WER, static thresholding yields approximately 3–11% lower RTF than dynamic thresholding.

4.3 Generalization Across DLM Backbones

To examine whether the observed decoding trends depend on the DLM backbone, we further compare LLaDA-8B (Nie et al., 2025) and Dream-7B (Ye et al., 2025) with a block size of 32 on LibriSpeech test-clean and test-other. As shown in Table 2, similar trends are observed across both backbones: threshold-based strategies yield lower WER than fixed-number decoding, while static and dynamic thresholding achieve nearly matched WER. At comparable WER, static thresholding yields 6.7–16.7% lower RTF than dynamic thresholding across the two backbones and test sets.

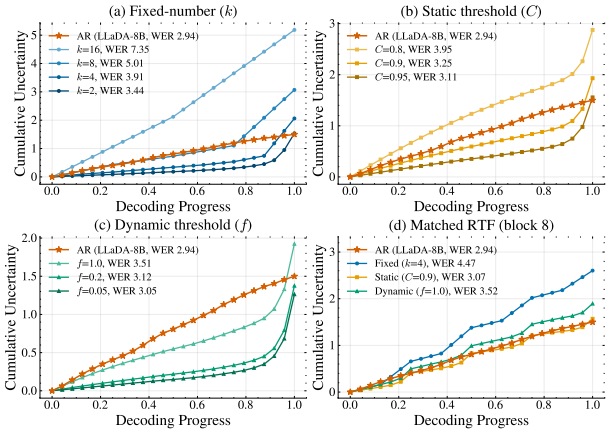


Figure 2: Cumulative uncertainty trajectories over decoding progress. Panels (a–c) compare each strategy with the AR reference; (d) compares all strategies under matched RTF.

4.4 Round-wise Progress Assessment

For this analysis, we normalize progress over the first 32 text-token positions and consider utterances with at least 32 text tokens, setting $P_r = 1$ when all 32 positions are fully unmasked.

4.4.1 Prediction Reliability using Uncertainty Trajectory

Measuring round-wise WER is impractical because insertion and deletion errors break alignment across decoding rounds. We therefore use cumulative uncertainty as a proxy for prediction reliability. As shown in Fig. 2(a-c), a larger final gap from the AR reference generally corresponds to higher WER, consistent with an association between cumulative uncertainty and recognition error. Most trajectories steepen near the end, suggesting that later rounds commit more low-confidence tokens; early deviation above the AR trajectory may reflect higher uncertainty in early commitments.

Under matched RTF (Fig. 2(d)), static-threshold decoding tracks the AR trajectory most closely and achieves the lowest WER among the compared configurations: 3.07% versus 4.47% for fixed-number and 3.52% for dynamic thresholding.

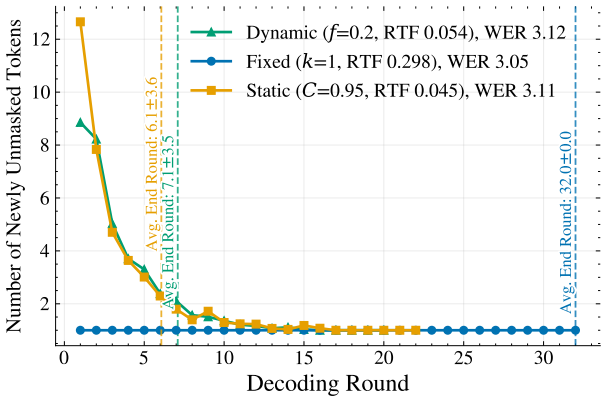


Figure 3: Per-round token throughput under comparable-WER settings. Dashed lines indicate mean stopping rounds.

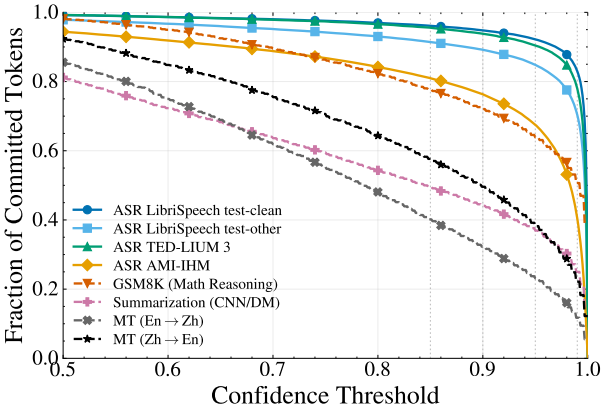


Figure 4: Complementary cumulative distribution function (CCDF) of token confidence scores. The plot shows the fraction of tokens with a confidence score S greater than or equal to a threshold s.

4.4.2 Inference Speed using Number of Unmasked Tokens

To examine where speed differences occur, we compare per-round token throughput under comparable WERs (3.05%–3.12%): static thresholding ($C = 0.95$), dynamic thresholding ($f = 0.2$), and fixed-number decoding ($k = 1$, included as a comparable-WER operating point). As shown in Fig. 3, static thresholding finishes in the fewest rounds on average (6.1), followed by dynamic thresholding (7.1) and fixed-number decoding (32.0), consistent with their RTFs of 0.045, 0.054, and 0.298, respectively.

The observed difference is concentrated in the early decoding rounds. Static thresholding exhibits a heavy-start pattern, committing many tokens early before tapering to final refinement, whereas dynamic thresholding proceeds more steadily and fixed-number decoding is constrained by its per-round budget.

4.5 Confidence Distribution and Decoding Efficiency

To examine the confidence distribution associated with the early commitment behavior, we compare commit-time confidence across ASR and several text tasks by committing the highest-confidence token at each step. The text tasks are evaluated using the base LLaDA-8B-Instruct model.

As shown in Fig. 4, confidence is generally more concentrated at high values in the evaluated ASR tasks than in the evaluated text tasks. At a confidence threshold of 0.90, 94.8% and 89.2% of tokens exceed the threshold on LibriSpeech test-clean and test-other, respectively, compared with 72.0% on GSM8K (Cobbe et al., 2021) and 44.1% on CNN/DailyMail (Hermann et al., 2015), with lower fractions observed for WMT19 machine translation (Barrault et al., 2019). TED-LIUM 3 remains close to LibriSpeech, while AMI-IHM shows a larger decrease but still retains more high-confidence tokens than all evaluated text tasks at thresholds from 0.85 to 0.95.

This confidence concentration is consistent with the observed differences in decoding efficiency. Static thresholding can commit many high-confidence tokens at once, whereas the dynamic-threshold criterion becomes more restrictive as more tokens are selected: $c^{(k)} > 1 - f/(k+1)$ requires $c^{(k)}$ to approach one as k increases. Fixed-number decoding, in contrast, is limited by its fixed per-round token budget.

5 Conclusion

We compared fixed-number, static-threshold, and dynamic-threshold decoding for DLM-based ASR across multiple English speech test sets and two DLM backbones. Across the evaluated settings, threshold-based decoding provides a better accuracy–speed trade-off than fixed-number decoding. Static and dynamic thresholding achieve comparable WER, with static thresholding generally yielding lower RTF. In the evaluated ASR settings, confidence scores are more concentrated at high values than in the examined text tasks, which is consistent with the observed early commitment behavior. The decoding-strategy trends are similar across the evaluated datasets and two DLM backbones, while confidence concentration decreases under domain shift. Future DLM decoding strategies for ASR may benefit from reducing late-stage refinement cost while maintaining recognition accuracy.

Limitations

We do not evaluate accented speech, low-resource or multilingual ASR, or ASR models trained directly on spontaneous speech. Although we evaluate two DLM backbones, all systems are trained only on LibriSpeech 960h. Results on TED-LIUM 3 and AMI-IHM therefore reflect domain shift for LibriSpeech-trained models rather than performance after in-domain training on prepared talks or spontaneous meetings. Confidence concentration also weakens under domain shift and is lower on AMI-IHM than on the LibriSpeech test sets. Whether the observed pattern holds for models trained on spontaneous speech or under broader linguistic and acoustic conditions remains an open question.

Ethical Considerations

This work uses publicly available datasets, pretrained models, and benchmarks, including LibriSpeech, TED-LIUM 3, AMI, WMT19, CNN/DailyMail, GSM8K, Whisper, LLaDA-8B-Instruct, Dream-7B, and LLaMA-3.1-8B. We use these resources solely for research purposes and follow their respective licenses, access conditions, and terms of use. We do not collect new human-subject data or conduct user studies. ASR systems may produce transcription errors and should be deployed with appropriate validation and human oversight, particularly in high-stakes settings.

Acknowledgments

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2022-NR070162), and in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by Korean Government (Ministry of Science and ICT) under Grant RS-2022-II220984.

References

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenhang Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, K. Lu, and 31 others. 2023. Qwen technical report. *ArXiv*, abs/2309.16609.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas S Nolte, and Brian Karrer. 2025. Accelerated sampling from masked diffusion models via entropy bounded unmasking. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 55981–56007. Curran Associates, Inc.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner. 2006. The ami meeting corpus: A pre-announcement. In *Machine Learning for Multimodal Interaction*, pages 28–39, Berlin, Heidelberg. Springer Berlin Heidelberg.

William Chan, Chitwan Saharia, Geoffrey Hinton, Mohammad Norouzi, and Navdeep Jaitly. 2020. Imputer: Sequence modelling via imputation and dynamic programming. In *International Conference on Machine Learning*, pages 1403–1413. PMLR.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. 2022. Maskgit: Masked generative image transformer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11305–11315.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Zhehuai Chen, He Huang, Andrei Andrusenko, Oleksii Hrinchuk, Krishna C Puvvada, Jason Li, Subhankar Ghosh, Jagadeesh Balam, and Boris Ginsburg. 2024. Salm: Speech-augmented language model with in-context learning for speech recognition and translation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 13521–13525. IEEE.

Ethan A Chi, Julian Salazar, and Katrin Kirchhoff. 2021. Align-refine: Non-autoregressive speech recognition via iterative realignment. In *Proceedings of the 2021*

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1920–1927.

Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. 2023. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Hengyu Fu, Baihe Huang, Virginia Adams, Charles Wang, Junkeun Yi, Mohammad Mahdi Kamani, Venkat Krishna Srinivasan, and Jiantao Jiao. 2026. From bits to rounds: Parallel decoding with exploration for diffusion language models. In *Forty-third International Conference on Machine Learning*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Esteve. 2018. Tedlium 3: Twice as much data and corpus repartition for experiments on speaker adaptation. In *International conference on speech and computer*, pages 198–208. Springer.

Yosuke Higuchi, Shinji Watanabe, Nanxin Chen, Tetsuji Ogawa, and Tetsunori Kobayashi. 2020. Mask CTC: Non-Autoregressive End-to-End ASR with CTC and Mask Predict. In *Interspeech 2020*, pages 3655–3659.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. Large language diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 50608–50646. Curran Associates, Inc.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Yifan Peng, Jinchuan Tian, William Chen, Siddhant Arora, Brian Yan, Yui Sudo, Muhammad Shakeel, Kwanghee Choi, Jiatong Shi, Xuankai Chang, Jee weon Jung, and Shinji Watanabe. 2024. OWSM v3.1: Better and Faster Open Whisper-Style Speech Models based on E-Branchformer. In *Interspeech 2024*, pages 352–356.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, Alexei Baevski, Yossi Adi, Xiaohui Zhang, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2024. Scaling speech technology to 1,000+ languages. *Journal of Machine Learning Research*, 25(97):1–52.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. In *Proc. Interspeech 2019*, pages 3465–3469.

Wenjie Tian, Bingshen Mu, Guobin Ma, Xuelong Geng, Zhixian Zhao, and Lei Xie. 2026. dllm-asr: A faster diffusion llm-based framework for speech recognition. *arXiv preprint arXiv:2601.17902*.

Mengqi Wang, Zhan Liu, Zengrui Jin, Guangzhi Sun, Chao Zhang, and Philip C Woodland. 2026. Audio-conditioned diffusion llms for asr and deliberation processing. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 18467–18471. IEEE.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. 2026. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. In *International Conference on Learning Representations*, volume 2026, pages 57027–57051.

Jian Wu, Yashesh Gaur, Zhuo Chen, Long Zhou, Yimeng Zhu, Tianrui Wang, Jinyu Li, Shujie Liu, Bo Ren, Linquan Liu, and Yu Wu. 2023. On decoder-only architecture for speech-to-text and large language model integration. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. 2025. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*.

Runpeng Yu, Xinyin Ma, and Xinchao Wang. 2025. Dimple: Discrete diffusion multimodal large language model with parallel decoding. *arXiv preprint arXiv:2505.16990*.

Setting	Configuration
Model	
Speech encoder	Whisper-medium.en (frozen)
Adapter	Conv1d ($k = 4$, $s = 4$) + Linear $\rightarrow$ 4096
Decoder	LLaDA-8B-Instruct
LoRA	{q, k, v, o}, $r = 16$, $\alpha = 32$, dropout 0.05
Answer canvas	128 tokens; EOS padding; left-padded batches
Training	
Objective	Cross-entropy
Optimizer	AdamW; lr $= 10^{-4}$; $\beta_1 = 0.9$, $\beta_2 = 0.999$; wd $= 0.01$
Schedule	Cosine decay; 500 warm-up; 100K steps
Batching	$\leq$12 utterances/GPU; gradient accumulation 2
Precision	bf16; DeepSpeed ZeRO-2; gradient clipping 1.0
Data and Compute	
Training data	LibriSpeech 960h
Hardware	8$\times$ NVIDIA RTX A6000
Training time	$\approx$25.8 h
Seed	10086

Table 3: Training configuration of the ASR system.

	k						
B	1	2	4	8	16	32	128
4	2.79/.205	3.62/.113	4.80/.067	–	–	–	–
8	2.93/.209	3.83/.111	4.47/.062	6.70/.038	–	–	–
16	3.01/.242	3.47/.122	4.02/.068	5.68/.038	9.07/.024	–	–
32	3.05/.298	3.44/.154	3.91/.080	5.01/.046	7.35/.028	12.10/.019	–
64	3.32/.489	3.57/.228	4.03/.114	4.80/.064	6.50/.037	10.50/.022	–
128	4.23/.627	4.23/.301	4.75/.148	5.42/.084	6.64/.052	9.92/.033	14.50/.014

Table 4: Fixed-number decoding on LibriSpeech test-clean. Each cell reports WER (%) / RTF. Dashes denote inapplicable settings.

A Experimental Details

A.1 Model and training.

Table 3 provides the full training configuration. Unless otherwise stated, all reported results are obtained from a single run with seed 10086. The Dream-7B variant in Section 4.3 replaces only the decoder, with all other settings unchanged.

A.2 Datasets.

All models are trained on LibriSpeech 960h. Evaluation is conducted on LibriSpeech test-clean (2,620 utterances, 5.4 h), test-other (2,939 utterances, 5.3 h), TED-LIUM 3 test (1,154 utterances, 2.6 h), and AMI-IHM test (12,643 segments, 8.7 h). TED-LIUM 3 and AMI-IHM are evaluated using the LibriSpeech-trained model without adaptation.

	C		
B	0.8	0.9	0.95
4	3.17/.081	2.85/.086	2.81/.100
8	3.53/.053	3.07/.058	2.96/.066
16	3.72/.039	3.15/.047	3.04/.052
32	3.95/.031	3.25/.038	3.11/.045
64	4.37/.029	3.62/.035	3.50/.043
128	4.93/.029	4.28/.038	4.13/.046

Table 5: Static thresholding on LibriSpeech test-clean. Each cell reports WER (%) / RTF.

	f		
B	1.0	0.2	0.05
4	3.32/.081	2.81/.093	2.79/.110
8	3.52/.059	2.97/.075	2.93/.096
16	3.41/.044	3.03/.061	3.02/.083
32	3.51/.039	3.12/.054	3.05/.081
64	3.88/.040	3.44/.061	3.31/.082
128	4.47/.042	4.18/.063	4.23/.089

Table 6: Dynamic thresholding on LibriSpeech test-clean. Each cell reports WER (%) / RTF.

A.3 Inference and latency measurement.

Decoding uses the dual prefix–suffix KV-cache of Fast-dLLM with a fixed 128-token answer canvas. Once an EOS token is committed, all remaining positions are filled with EOS and subsequent blocks are skipped. All latency measurements are performed on a single idle NVIDIA RTX A6000 with batch size 1 and one evaluation job per GPU. RTF is computed as the total inference time, including the speech encoder, divided by the total audio duration. Average NFE is the mean number of decoder forward passes per utterance.

A.4 AR baselines.

AR (LLaDA-8B) decodes the same fine-tuned DLM one token at a time from left to right and serves as the reliability reference. AR (LLaMA-3.1-8B) is a LoRA-tuned autoregressive decoder trained with the same speech encoder, adapter, training data, and optimization schedule.

B Additional Results and Analyses

B.1 Full Accuracy–Speed Sweep on LibriSpeech test-clean

Tables 4–6 report the numerical values underlying Fig. 1. Each table covers one commitment strategy over its full hyperparameter range and all six block sizes, with WER (%) and RTF reported for each

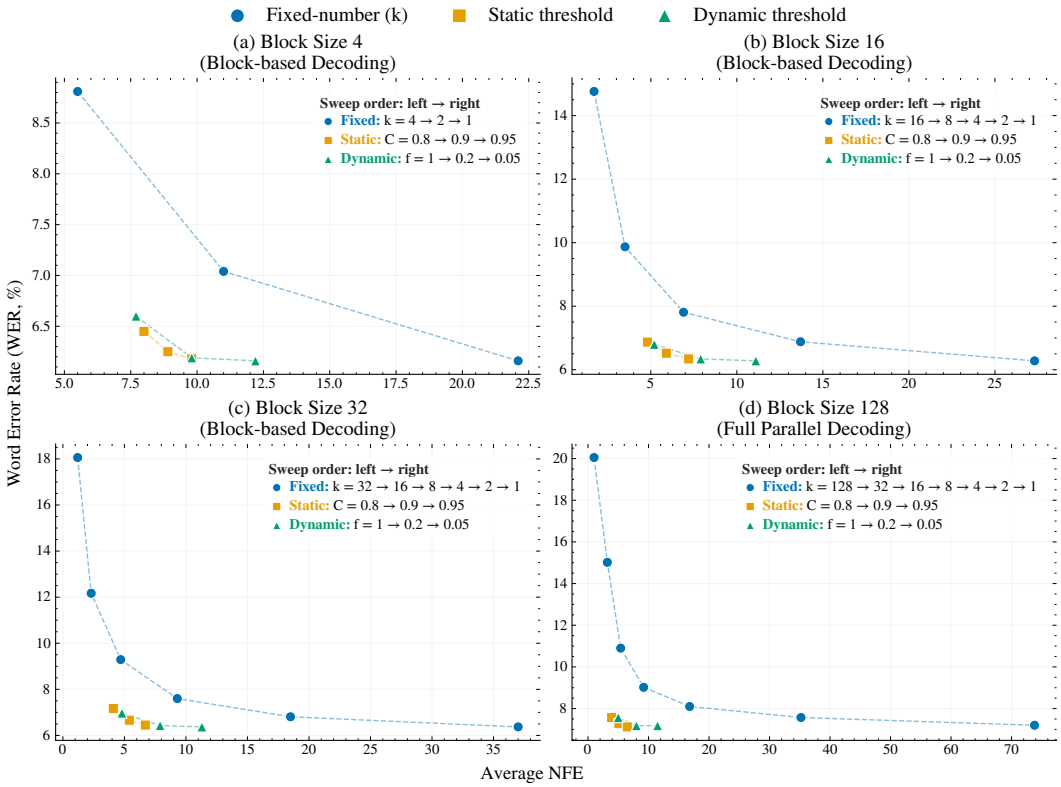


Figure 5: WER–NFE trade-off on LibriSpeech test-other across decoding strategies and block sizes. Parameters in each panel are listed in order of increasing NFE. Lower WER and NFE are preferred.

	k						
B	1	2	4	8	16	32	128
4	6.16/22.1	7.04/11.0	8.81/5.5	–	–	–	–
16	6.28/27.3	6.88/13.7	7.81/6.9	9.87/3.5	14.76/1.7	–	–
32	6.37/37.0	6.81/18.5	7.60/9.3	9.29/4.7	12.17/2.3	18.06/1.2	–
128	7.20/73.8	7.57/35.2	8.10/16.8	9.02/9.2	10.90/5.4	15.02/3.2	20.05/1.0

Table 7: Fixed-number decoding on LibriSpeech test-other. Each cell reports WER (%) / average NFE. Dashes denote inapplicable settings.

	C		
B	0.8	0.9	0.95
4	6.45/8.0	6.25/8.9	6.18/9.8
16	6.87/4.8	6.52/5.9	6.34/7.2
32	7.17/4.1	6.66/5.4	6.45/6.7
128	7.57/3.9	7.28/5.1	7.12/6.5

Table 8: Static thresholding on LibriSpeech test-other. Each cell reports WER (%) / average NFE.

configuration. The AR reference points are independent of block size: AR (LLaDA-8B) achieves 2.94% WER at RTF 0.276, while AR (LLaMA-3.1-8B) achieves 2.78% WER at RTF 0.160. Across block sizes up to 64, stricter commitment criteria consistently yield lower WER within each strategy.

	f		
B	1.0	0.2	0.05
4	6.60/7.7	6.19/9.8	6.16/12.2
16	6.80/5.2	6.34/7.9	6.28/11.1
32	6.97/4.8	6.43/7.9	6.37/11.3
128	7.57/5.0	7.18/8.0	7.18/11.5

Table 9: Dynamic thresholding on LibriSpeech test-other. Each cell reports WER (%) / average NFE.

B.2 Full Accuracy–NFE Sweep on LibriSpeech test-other

Tables 7–9 report the corresponding sweep on LibriSpeech test-other for block sizes $B \in \{4, 16, 32, 128\}$, and Fig. 5 shows the resulting trade-off curves. Each cell reports WER (%) / average NFE, using NFE as a hardware-independent measure of decoding cost. RTF values for representative test-other configurations are reported in Table 1. The overall ordering observed on test-clean is preserved: threshold-based decoding consistently provides lower WER than fixed-number decoding at comparable decoding cost, while static and dynamic thresholding remain closely matched.

Test set	Backbone	Static	Dynamic	Fixed-k
Table 1				
test-other	–	7.2	7.9	13.7
TED-LIUM 3	–	7.0	8.0	16.8
AMI-IHM	–	6.7	7.1	9.2
Table 2				
test-clean	LLaDA-8B	5.0	6.2	9.8
	Dream-7B	7.0	8.1	10.8
test-other	LLaDA-8B	6.7	7.9	9.3
	Dream-7B	8.7	9.6	10.4

Table 10: Average NFE per utterance for the main-text configurations in Tables 1 and 2.

Model	WER (%)		RTF	
	clean	other	clean	other
Whisper-medium.en (greedy)	3.1†	6.3†	0.078	0.079
Whisper-medium.en (beam 5)	3.0†	5.7†	0.085	0.087
Whisper-large-v3 (greedy)	2.17	4.05	0.087	0.103
Whisper-large-v3 (beam 5)	2.00	3.71	0.125	0.128
Ours (static C=0.95, B=4)	2.81	6.18	0.100	0.107
Ours (static C=0.95, B=16)	3.04	6.34	0.052	0.072

Table 11: Comparison with Whisper baselines on LibriSpeech. WERs marked with † are taken from Radford et al. (2023). All other WERs and all RTF values were measured by us on a single idle NVIDIA RTX A6000 with batch size 1. Our block-16 configuration reaches WER close to Whisper-medium.en greedy decoding at a lower RTF, while Whisper-large-v3 attains lower WER at a higher RTF.

B.3 Average NFE for Main-Text Configurations

Tables 1 and 2 report RTF as the primary cost metric because it reflects wall-clock latency on the target hardware. For completeness, Table 10 reports the average number of decoder forward passes (NFE) per utterance for the same configurations. NFE follows the same overall efficiency trend as RTF: static thresholding uses the fewest forward passes in every setting, while dynamic thresholding requires 0.4–1.2 additional passes per utterance at nearly matched WER. Fixed-number decoding requires more forward passes than static thresholding while also yielding higher WER.

B.4 Comparison with Whisper Baselines

Table 11 compares our system with Whisper-medium.en and Whisper-large-v3. All RTF values are measured under the same latency protocol. On LibriSpeech test-clean, our block-16 static-threshold configuration achieves WER close to Whisper-medium.en greedy decoding at lower RTF, while Whisper-large-v3 achieves lower WER at higher RTF. This comparison is intended to contextualize the absolute performance rather than claim state-of-the-art accuracy, as our system uses the Whisper-medium.en encoder and is trained only on LibriSpeech 960h.

Task	N	$\geq$0.85	$\geq$0.90	$\geq$0.95	$\geq$0.99
test-clean	50,046	96.1	94.8	92.2	83.0
test-other	51,514	91.4	89.2	84.8	70.4
TED-LIUM 3	25,195	95.6	93.8	90.4	78.0
AMI-IHM	83,000	81.0	76.3	67.5	40.4
GSM8K	16,000	77.7	72.0	63.9	50.7
CNN/DailyMail	15,999	49.6	44.1	36.9	25.0
MT (En→Zh)	9,667	40.2	32.2	23.1	11.7
MT (Zh→En)	10,854	57.6	49.6	38.2	22.2

Table 12: Fraction (%) of committed tokens with confidence at or above each threshold under sequential confidence-ordered decoding (block size 32, first 32-token block preceding the first EOS, EOS excluded). ASR tasks are listed above the rule and text tasks below it.

B.5 Confidence Statistics Across ASR and Text Tasks

Table 12 reports the statistics underlying Fig. 4. Following Section 4.5, we use sequential confidence-ordered decoding with block size 32, committing only the highest-confidence token at each step. Commit-time confidence is recorded for positions in the first 32-token block preceding the first EOS token, with EOS excluded. The text tasks use the base LLaDA-8B-Instruct model with its chat template on 500 examples each from GSM8K, CNN/DailyMail, and WMT19 in both translation directions. The ASR tasks use the LibriSpeech-trained model on the full test sets. N denotes the number of pooled token commitments per task.

Across thresholds from 0.85 to 0.95, every evaluated ASR dataset retains a larger fraction of high-confidence tokens than any of the text tasks. AMI-IHM shows the largest decrease among the ASR datasets under domain shift and falls below GSM8K at the 0.99 threshold, indicating that confidence concentration at the extreme upper end is more sensitive to acoustic conditions.

B.6 Statistical Validation

Utterance-level NLL–WER association. To statistically assess the relationship between cumulative uncertainty and recognition error discussed in Fig. 2, we compute the Spearman rank corre-

lation between the final cumulative committed-token NLL and utterance-level WER under dynamic thresholding ($f = 1.0$, block size 32). The correlation is $\rho = 0.627$ on LibriSpeech test-clean ($n = 2{,}620$) and $\rho = 0.672$ on test-other ($n = 2{,}939$). These results support a positive utterance-level association between cumulative uncertainty and recognition error: utterances with larger accumulated uncertainty tend to have higher WER.

Because cumulative committed-token NLL reflects the model's confidence in its own predictions rather than prediction correctness, it need not track WER uniformly across decoding configurations. This limitation can become relevant in near-single-shot fully parallel settings, where many tokens are committed from a largely masked context. We therefore use the per-utterance correlations above as the primary statistical evidence supporting the uncertainty-trajectory analysis, rather than drawing conclusions from an across-configuration correlation.

Paired WER comparisons on LibriSpeech test-other. We further assess the WER differences between static-threshold and fixed-number decoding using a paired utterance-level bootstrap with 10,000 resamples on LibriSpeech test-other with block size 16. For the configurations reported in Table 1, static thresholding ($C = 0.95$) reduces WER relative to fixed-number decoding with $k = 2$ by 0.54 absolute percentage points, with a 95% confidence interval of [0.44, 0.64] and $p < 10^{-4}$. The average NFE is 7.2 for static thresholding and 13.7 for fixed-number decoding.

We additionally compare static thresholding against fixed-number decoding with $k = 4$, which approximately matches its decoding computation (7.2 versus 6.9 average NFE). At this matched-computation operating point, static thresholding reduces WER by 1.47 absolute percentage points, with a 95% confidence interval of [1.25, 1.65] and $p < 10^{-4}$. These results indicate that the WER differences between static thresholding and the tested fixed-number configurations are statistically significant both for the main-text configuration and under approximately matched decoding computation. We restrict claims of statistical significance to these two tested comparisons.