Training-Free Character-Length Control in Summarization with Diffusion Language Models

Steffen Castle[1] **Nils Feldhus**[3,1] **Christopher Ebert**[1]
Leonhard Hennig[1] **Sebastian Möller**[1,2]

[1]German Research Center for Artificial Intelligence (DFKI)
[2]Technische Universität Berlin [3]CLCG, University of Groningen

Abstract

Summarization with a fixed character length is important for tasks such as headline and article preview generation, where text must fit an exact space. Autoregressive language models (LMs) struggle with precise character-length control: training-based methods require task-specific fine-tuning, while zero-shot prompting strategies achieve only approximate compliance. We propose DELTA (**D**iffusion **E**xpected **L**ength **T**argeting **A**lgorithm), a training-free inference-time method for masked diffusion LMs that estimates expected character length at each denoising step and dynamically inserts or removes masked positions to steer toward a target length. DELTA requires no fine-tuning and is designed for off-the-shelf masked diffusion LMs. Across three summarization benchmarks, it reduces character-length errors by an order of magnitude compared to baselines while maintaining competitive summary quality in both automatic and human evaluation.

1 Introduction

Many summarization tasks require outputs that fit a precise character budget. In publishing workflows, for example, headlines and article previews are often produced for predefined display slots with fixed character capacities (Hitomi et al., 2019); similar constraints arise in notification previews, social media posts, subtitles, and UI text, where even a few extra characters can cause truncation or layout overflow. Existing approaches to length-controlled summarization operate within the constraints of autoregressive generation: training-based methods (Miculicich et al., 2023; Belligoli et al., 2025) modify the model or its objective to encourage a target length, while zero-shot strategies (Retkowski and Waibel, 2025; Xie and Lee, 2025) rely on prompting techniques or post-hoc filtering. Both approaches operate under standard left-to-right autoregressive decoding, where previously emitted tokens are not directly revised within the same generation pass. This makes it difficult to dynamically correct the decoded character length as generation proceeds.

Masked diffusion language models such as LLaDA (Nie et al., 2025) and Dream (Ye et al., 2025) offer a structurally different generation process. These models iteratively refine a buffer of masked tokens over multiple denoising steps, and are trained to generate across a range of output lengths. This means that occasional insertions or removals of mask tokens will not severely disrupt generation quality. We exploit this property with DELTA, an inference-time method that estimates expected character length from both committed tokens and the probability distributions over remaining masks, then adjusts the buffer to steer toward a target. DELTA requires no fine-tuning or architectural changes and is designed to apply to off-the-shelf masked diffusion LMs. Across three summarization benchmarks, it reduces character-length errors by an order of magnitude compared to all baselines while maintaining competitive summary quality.

2 Background and Related Work

Length-Controlled Summarization. Prior work on length-controlled summarization predominantly operates within the autoregressive framework and spans token-, word-, sentence-, and character-level constraints. Training-based approaches modify either the model or its objective: CTRLsum (He et al., 2020) conditions generation on coarse length bins, Miculicich et al. (2023) jointly train a length predictor and constrained decoder for precise token or sentence counts, and Belligoli et al. (2025) upweight the EOS-token loss to enforce a character-length ceiling.

Training-free approaches instead steer pretrained autoregressive LMs at inference time. Retkowski

and Waibel (2025) convert character targets to word targets using statistical ratios and filter multiple samples, while Xie and Lee (2025) append countdown markers to obtain near-exact word-level control. Instruction- and position-based methods likewise target explicit length constraints, including sentence compression with InstructCMP (Juseon-Do et al., 2024) and positional representations for length control (Wang et al., 2024). Other inference-time approaches address precise length control in LLMs (Butcher et al., 2025), black-box length-constrained generation (Gu et al., 2025), and iterative controllable summarization (Ryu et al., 2026).

Controllable generation has also been studied in diffusion language models, including methods that steer generation toward desired attributes (Li et al., 2022). In contrast to these approaches, DELTA exploits masked-diffusion denoising to directly estimate and dynamically correct decoded character length during generation, requiring neither model training nor post-hoc regeneration.

Discrete Diffusion Language Models. D3PM (Austin et al., 2021) introduced discrete diffusion over token vocabularies, and MDLM (Sahoo et al., 2024) refined the training objective with a Rao-Blackwellized loss that substantially narrows the likelihood gap with autoregressive models. These foundations enabled scaling to billion parameters: LLaDA (Nie et al., 2025) trains an 8B masked diffusion model from scratch, competitive with LLaMA 3, while Dream (Ye et al., 2025) initializes from a pretrained autoregressive checkpoint with context-adaptive noise rescheduling. Both generate output by iteratively unmasking tokens using confidence-based strategies originating in MaskGIT (Chang et al., 2022).

Variable-Length Generation in Diffusion LMs. Standard masked diffusion decoding requires choosing a fixed token-buffer size before denoising. This token-buffer length is distinct from a target decoded character length because tokens correspond to strings of varying character length. DreamOn (Wu et al., 2026) introduces special expand/delete tokens that allow the model to adjust the token-buffer length during denoising, requiring fine-tuning. DAEDAL (Li et al., 2026) and ρ-EOS (Yang et al., 2026) are training-free alternatives that adapt sequence length during open-ended generation. These methods address variable-length generation rather than generation toward a specified decoded character budget.

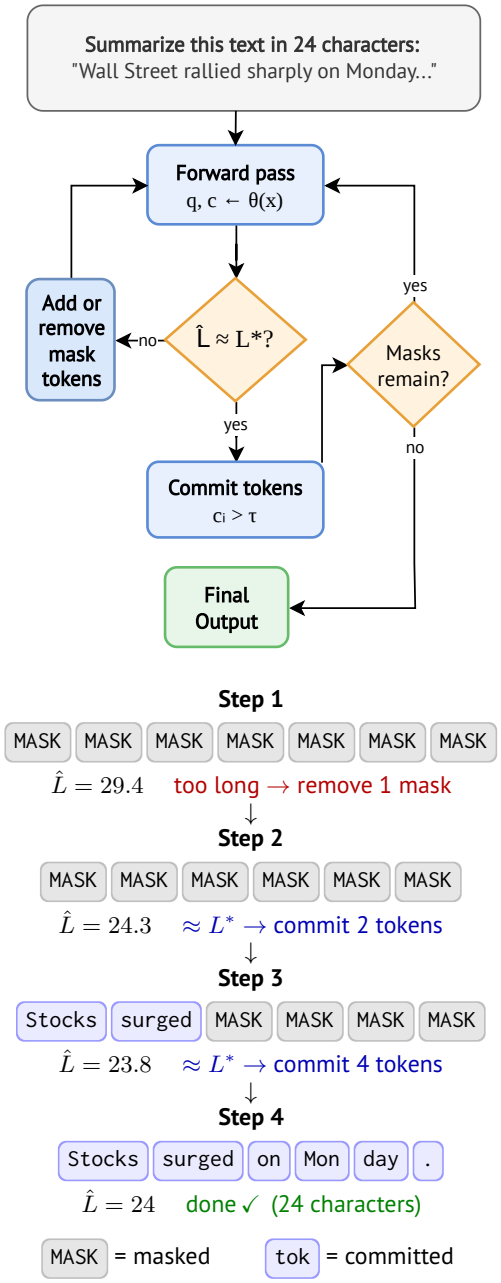


Figure 1: Overview of DELTA. **Top:** Algorithm flowchart. **Bottom:** Example with target $L^* = 24$.

3 Methodology: DELTA

We describe DELTA (**D**iffusion **E**xpected **L**ength **T**argeting **A**lgorithm), an inference-time method that steers masked diffusion language models toward a target character length. DELTA operates within the diffusion iterative denoising loop and requires no fine-tuning or architectural changes.

3.1 Preliminaries

Let V denote the vocabulary and $\ell : V \to \mathbb{N}_0$ the character-length function, which maps each token to the number of characters in its decoded string

(with $\ell(v) = 0$ for special tokens such as [MASK], [PAD], and [EOT]). Because tokens may decode to strings of different character lengths, fixing the number of output tokens does not in general fix the decoded character length. Given a prompt $\mathbf{p} = (p_1, \ldots, p_n)$, a masked diffusion LM generates by initializing an output buffer $\mathbf{x} = (x_1, \ldots, x_m)$ with all positions set to [MASK] and iteratively predicting tokens at masked positions until none remain.

At each denoising step, the model produces a probability distribution $\mathbf{q}_i \in \Delta^{|V|}$ over the vocabulary for each output position i. These distributions, combined with the character-length function ℓ, provide an estimate of the output's character length at every point during generation, even before any tokens are committed.

3.2 Expected Character Length

We define the *expected character length* of the current output as $\hat{L} = \sum_{i=1}^{m} \hat{\ell}_i$, where $\hat{\ell}_i$ depends on whether position i has been committed (unmasked) or is still masked:

$$\hat{\ell}_i = \begin{cases} \ell(x_i) & \text{if } x_i \neq \texttt{[MASK]}, \\ \sum_{v \in V} q_{i,v} \cdot \ell(v) & \text{if } x_i = \texttt{[MASK]}. \end{cases} \quad (1)$$

For committed tokens, $\hat{\ell}_i$ is the exact character length. For masked positions, $\hat{\ell}_i$ is the probability-weighted expected character length under the model's current prediction. We additionally mask out zero-length (special) tokens from logits before computing $\mathbf{q}_i$, ensuring they are never predicted.

3.3 Confidence Scoring

At each step, the algorithm must decide which masked positions to unmask. Following MaskGIT-style decoding (Chang et al., 2022), we assign a confidence score c_i to each position and unmask those exceeding a threshold τ. We use the maximum predicted probability as the confidence score: $c_i = \max_{v \in V} q_{i,v}$. At each step, at least one position is always unmasked (the one with highest confidence), guaranteeing progress.

3.4 Length-Controlled Denoising

The algorithm combines three operations: *length estimation*, *buffer adjustment*, and *token commitment*. A schematic is shown in Algorithm 1.

Buffer adjustment. After computing $\hat{L}$ and the confidence scores, the algorithm compares $\hat{L}$ to the target L^*. If $\hat{L} > L^*$, the buffer is too long and masked positions should be removed. We sort masked positions by ascending confidence (least confident first) and greedily remove masks until the expected length deficit is minimized:

$$n_{\text{remove}} = \arg\min_{k} \left| \hat{L} - \sum_{j=1}^{k} \hat{\ell}_{\sigma(j)} - L^* \right|, \quad (2)$$

where σ is the sorted ordering. Removal is restricted to masks whose confidence falls below a threshold τ_{low}, preventing deletion of positions the model is already confident about. If instead $\hat{L} < L^*$, new [MASK] tokens are inserted adjacent to the least-confident masked positions. The number of insertions is estimated as $(L^* - \hat{L})/\bar{\ell}$, where $\bar{\ell}$ is the current mean expected character length per token.

Token commitment. After any buffer adjustment, masked positions with confidence $c_i > \tau$ are unmasked by replacing them with their argmax predictions. To prevent oscillation, where the algorithm alternates between inserting and removing masks without making progress, buffer adjustments are only permitted immediately after a commitment step. Since each commitment step unmasks at least

Algorithm 1 Length-Controlled Diffusion Decoding

Require: Diffusion LM θ, prompt $\mathbf{p}$, target length L^*, initial buffer size m, confidence threshold τ

$\mathbf{x} \leftarrow [\mathbf{p}; \texttt{MASK}^m; \texttt{EOT}]$
can_adjust $\leftarrow$ true
while $\mathbf{x}$ contains MASK tokens **do**
  $\mathbf{q}, \mathbf{c} \leftarrow \text{ForwardPass}(\theta, \mathbf{x})$
  $\hat{L} \leftarrow \text{ExpectedCharLength}(\mathbf{x}, \mathbf{q})$
  if $\hat{L} > L^*$ **and** *can_adjust* **then**
    Remove lowest-confidence masks to reduce $\hat{L}$ toward L^*
    can_adjust $\leftarrow$ false
  else if $\hat{L} < L^*$ **and** *can_adjust* **then**
    Insert new masks at lowest-confidence positions
    can_adjust $\leftarrow$ false
  else
    Unmask positions where $c_i > \tau$ (at least one)
    can_adjust $\leftarrow$ true
  end if
end while
return $\mathbf{x}$

Method	Length		Quality			G-Eval			
	MAE	<10	R-1	R-L	BF1	Rel	Con	Coh	Flu
XSum									
DELTA+LLaDA	**2.3**	99.6	0.286	0.217	0.882	3.41	4.21	4.61	4.66
DELTA+Dream	**2.3**	**99.7**	**0.333**	**0.255**	**0.890**	3.17	4.14	4.76	4.82
LA-SF-AR	5.2	84.2	0.273	0.204	0.881	3.56	4.38	4.79	4.89
GPT-4o-mini	13.1	48.4	0.264	0.194	0.880	**3.83**	4.73	**4.96**	**4.99**
Lead-char-snap	15.7	60.5	0.166	0.122	0.850	1.80	4.72	3.08	3.31
TextRank	14.5	50.1	0.178	0.130	0.845	1.74	**4.79**	4.07	4.59
CNN/DailyMail									
DELTA+LLaDA	**2.4**	**99.6**	0.363	0.259	0.876	4.04	4.59	4.58	4.49
DELTA+Dream	3.0	99.3	0.409	0.281	0.880	3.97	4.72	4.75	4.73
LA-SF-AR	10.3	59.7	0.399	0.254	0.879	4.29	4.61	4.94	4.96
GPT-4o-mini	96.6	11.1	0.393	0.245	0.877	**4.53**	4.86	**4.99**	**5.00**
BART-CNN	104.7	5.7	**0.441**	**0.306**	**0.882**	3.69	4.83	4.60	4.61
Lead-char-snap	59.4	21.4	0.363	0.230	0.868	2.85	**4.92**	4.22	4.40
TextRank	17.3	32.4	0.356	0.235	0.866	2.40	4.80	3.38	4.07
arXiv									
DELTA+LLaDA	**2.8**	**98.4**	0.443	0.234	0.837	3.71	4.35	4.43	4.39
DELTA+Dream	3.1	98.3	0.431	0.236	**0.853**	3.09	4.42	3.60	3.78
LA-SF-AR	44.9	22.9	**0.466**	**0.250**	0.845	3.98	4.28	4.78	4.98
GPT-4o-mini	112.2	3.7	0.432	0.217	0.839	**4.35**	4.65	**4.97**	**5.00**
Lead-char-snap	96.5	7.0	0.351	0.183	0.837	2.22	**4.99**	4.11	4.27
TextRank	46.9	12.0	0.406	0.212	0.837	2.52	4.55	3.00	3.19

Table 1: Results on XSum, CNN/DailyMail, and arXiv, ordered by increasing reference summary length. MAE is mean absolute character-length error; <10 is the percentage of outputs within 10 characters of the target. R-1/R-L = ROUGE-1/L; BF1 = BERTScore F1. G-Eval scores (1–5 Likert) use GPT-5-mini as judge (Liu et al., 2023). Best values per column within each dataset are **bolded**. BART-CNN is included only for CNN/DailyMail as it was fine-tuned on that dataset.

one token, this guarantees that the sequence state changes between consecutive adjustments, breaking potential cycles. The algorithm terminates when no masked positions remain or a maximum iteration count is reached.

4 Evaluation

We evaluate on three datasets: CNN/DailyMail (Hermann et al., 2015), a news summarization benchmark; XSum (Narayan et al., 2018), which targets single-sentence summaries; and arXiv (Cohan et al., 2018), which contains scientific paper abstracts. For each reference summary, we set the target character length to the character length of the reference. Dataset statistics are provided in Appendix A.

Baselines. We compare DELTA with LLaDA-8B-Instruct (Nie et al., 2025) and Dream-7B-Instruct (Ye et al., 2025) against: TextRank (Mihalcea and Tarau, 2004); Lead-char-snap, which selects the maximum number of initial source sentences that fit within the target character budget; BART fine-tuned on CNN/DailyMail (Lewis et al., 2020); GPT-4o-mini prompted with the target character count; and LA-SF-AR (Retkowski and Waibel, 2025) using Llama-3.1-8B-Instruct. Full baseline and hyperparameter details are in Appendix I. We additionally evaluate DAEDAL (Li et al., 2026), a training-free variable-length diffusion decoding method, using the same character-target prompts as the other generative methods. Results are reported in Appendix D. To isolate the contribution of dynamic buffer adjustment, we also evaluate a fixed-buffer ablation using the same diffusion models, prompts, confidence schedules, and initialization as DELTA, but with mask insertion and removal disabled. Results are reported in Appendix C. [1]

Metrics. We report mean absolute character-length error (MAE) and the percentage of outputs within 10 characters of the target, along with summary quality via ROUGE-1, ROUGE-L (Lin, 2004), and BERTScore F1 (Zhang et al., 2020). We additionally evaluate using LLM-as-a-judge following the G-Eval framework (Liu et al., 2023), which

[1] Implementation, evaluation scripts, and experimental configuration are publicly available at https://github.com/DFKI-NLP/delta-diffusion.

scores summaries on *relevance* (Rel), *consistency* (Con), *coherence* (Coh), and *fluency* (Flu) on a 1–5 Likert scale. We use GPT-5-mini as the judge; we additionally report GPT-4o-mini scores in the Appendix. All prompts are listed in Appendix H.

Human evaluation. We additionally conduct a blind human evaluation on 50 CNN/DailyMail examples comparing DELTA+Dream, LA-SF-AR, and GPT-4o-mini. Two annotators independently rate anonymized and randomly ordered outputs on relevance, consistency, coherence, and fluency using the same 1–5 scale as the G-Eval evaluation. Full results and agreement statistics are reported in Appendix B.

Runtime. We report wall-clock inference time per summary for DELTA and the non-API baselines. Values are measured as mean $\pm$ standard deviation over generated summaries; full results are reported in Appendix E.

5 Results

Results are shown in Table 1. On all datasets, DELTA achieves the best length compliance by a wide margin. While baseline errors scale with summary length, DELTA maintains MAE below 3.1 characters even on arXiv, where references average 969 characters. LA-SF-AR, the strongest autoregressive baseline for length control, places 84.2% of outputs within 10 characters on XSum but only 22.9% on arXiv, whereas DELTA remains above 98% across all datasets.

On summary quality, DELTA remains competitive. On XSum, DELTA+Dream achieves the highest ROUGE and BERTScore values, while on arXiv it achieves the highest BERTScore. On CNN/DailyMail, DELTA+Dream is second only to the fine-tuned BART baseline on ROUGE and BERTScore. Autoregressive baselines generally receive higher GPT-5-mini G-Eval scores, but this advantage comes with substantially poorer character-length compliance. A complementary human evaluation on 50 CNN/DailyMail examples finds DELTA+Dream competitive with LA-SF-AR and GPT-4o-mini, with average ratings of 4.27, 4.32, and 4.31 respectively and 91.5% within-one-point annotator agreement; full results are reported in Appendix B. Additional analyses confirm that DELTA's length-control gains come from dynamic buffer adjustment. Disabling mask insertion and removal in the fixed-buffer ablation substantially worsens character-length compliance while leaving summary quality broadly similar. Likewise, DAEDAL produces reasonable summaries but does not reliably satisfy specified character budgets. Full results are reported in Appendices C and D.

6 Conclusion

We presented DELTA, a training-free inference-time method that steers masked diffusion language models toward exact character-length targets by estimating expected length from token distributions and dynamically adjusting the generation buffer. Applied to LLaDA and Dream, DELTA reduces character-length errors by an order of magnitude compared to strong baselines while maintaining competitive summary quality in both automatic and human evaluation. Future work could address inference speed, evaluate DELTA across languages and model scales, and extend the approach to other constrained generation tasks.

Limitations

DELTA has several limitations. First, inference is slow. Diffusion language models require multiple full-sequence forward passes during denoising and, unlike autoregressive models, cannot use standard KV caching due to their bidirectional attention (Nie et al., 2025). On XSum, DELTA requires 0.67–0.69 seconds per summary, compared with 1.43 seconds for LA-SF-AR. On CNN/DailyMail and arXiv, however, DELTA requires 8.59 and 20.75–22.80 seconds per summary, respectively, compared with 2.19 and 13.22 seconds for LA-SF-AR. Thus, the tighter character-length control comes with increased inference cost on longer inputs. Full runtime measurements are reported in Appendix E. Recent work on accelerating diffusion LM inference through approximate KV caching and guided decoding (Hu et al., 2025) may mitigate this cost.

Second, automatic quality evaluation remains sensitive to the choice of LLM judge. GPT-4o-mini and GPT-5-mini produce substantially different Consistency and Coherence scores for several methods, particularly extractive baselines, consistent with known limitations of LLM-based judges (Chehbouni et al., 2025). Our human evaluation provides additional evidence that DELTA's length control does not cause a substantial quality regression, but it is limited to 50 CNN/DailyMail examples and two annotators. Larger human evaluations

across datasets would provide stronger evidence.

Third, our evaluation is limited to English-language summarization and to two diffusion models in the 7–8B parameter range. Both LLaDA and Dream are predominantly trained on English data and perform poorly on other languages in our informal testing. The effectiveness of character-level length estimation also depends on the token-to-character mapping of the underlying tokenizer, which varies across languages and writing systems. Although DELTA operates on output distributions rather than model-specific internal components, its effectiveness on substantially smaller or larger diffusion models remains to be validated.

Finally, the hyperparameters τ and τ_{low} vary across datasets and models, and we selected them via grid search on validation sets. We do not provide a sensitivity analysis or adaptive method for setting these thresholds, which limits the plug-and-play applicability of DELTA to new domains without tuning.

Acknowledgments

This work was funded by the German Federal Ministry of Research, Technology and Space (BMFTR) under the GenKI4Media project (grant 01MK25004E).

References

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA. Curran Associates Inc.

Zeno Belligoli, Emmanouil Stergiadis, Eran Fainman, and Ilya Gusev. 2025. Controlling summarization length through eos token weighting. *Preprint*, arXiv:2506.05017.

Bradley Butcher, Michael O'Keefe, and James Titchener. 2025. Precise length control for large language models. *Natural Language Processing Journal*, 11:100143.

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

Khaoula Chehbouni, Mohammed Haddou, Jackie CK Cheung, and Golnoosh Farnadi. 2025. Neither valid nor reliable? investigating the use of LLMs as judges. In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems Position Paper Track*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Yuxuan Gu, Wenjie Wang, Xiaocheng Feng, Weihong Zhong, Kun Zhu, Lei Huang, Ting Liu, Bing Qin, and Tat-Seng Chua. 2025. Length controlled generation for black-box LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16878–16895, Vienna, Austria. Association for Computational Linguistics.

Junxian He, Wojciech Kryściński, Bryan McCann, Nazneen Rajani, and Caiming Xiong. 2020. Ctrl-sum: Towards generic controllable text summarization. *arXiv*.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA. MIT Press.

Yuta Hitomi, Yuya Taguchi, Hideaki Tamori, Ko Kikuta, Jiro Nishitoba, Naoaki Okazaki, Kentaro Inui, and Manabu Okumura. 2019. A large-scale multi-length headline corpus for analyzing length-constrained headline generation model evaluation. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 333–343, Tokyo, Japan. Association for Computational Linguistics.

Zhanqiu Hu, Jian Meng, Yash Akhauri, Mohamed S. Abdelfattah, Jae-sun Seo, Zhiru Zhang, and Udit Gupta. 2025. Flashdlm: Accelerating diffusion language model inference via efficient kv caching and guided diffusion. *arXiv preprint arXiv:2505.21467v2*.

Juseon-Do, Jingun Kwon, Hidetaka Kamigaito, and Manabu Okumura. 2024. InstructCMP: Length control in sentence compression through instruction-based large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8980–8996, Bangkok, Thailand. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jinsong Li, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Jiaqi Wang, and Dahua Lin. 2026. Beyond fixed: Training-free variable-length denoising for diffusion large language models. In *The Fourteenth International Conference on Learning Representations*.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. 2022. Diffusion-lm improves controllable text generation. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Lesly Miculicich, Yujia Xie, Song Wang, and Pengcheng He. 2023. Summarization with precise length control. *Preprint*, arXiv:2305.05171.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Fabian Retkowski and Alexander Waibel. 2025. Zero-shot strategies for length-controllable summarization. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 551–572, Albuquerque, New Mexico. Association for Computational Linguistics.

Sangwon Ryu, Heejin Do, Daehui Kim, Hwanjo Yu, Dongwoo Kim, Yunsu Kim, Gary Lee, and Jungseul Ok. 2026. Exploring iterative controllable summarization with large language models. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 512–528, Rabat, Morocco. Association for Computational Linguistics.

Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M Rush, Yair Schiff, Justin T Chiu, and Volodymyr Kuleshov. 2024. Simple and effective masked diffusion language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Noah Wang, Feiyu Duan, Yibo Zhang, Wangchunshu Zhou, Ke Xu, Wenhao Huang, and Jie Fu. 2024. PositionID: LLMs can control lengths, copy and paste with explicit positional awareness. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16877–16915, Miami, Florida, USA. Association for Computational Linguistics.

Zirui Wu, Lin Zheng, Zhihui Xie, Jiacheng Ye, Jiahui Gao, Shansan Gong, Yansong Feng, Zhenguo Li, Wei Bi, Guorui Zhou, and Lingpeng Kong. 2026. Dreamon: Diffusion language models for code infilling beyond fixed-size canvas. In *The Fourteenth International Conference on Learning Representations*.

Juncheng Xie and Hung-yi Lee. 2025. Prompt-based one-shot exact length-controlled generation with llms. *Preprint*, arXiv:2508.13805.

Jingyi Yang, Yuxian Jiang, and Jing Shao. 2026. ρ-EOS: Training-free bidirectional variable-length control for masked diffusion llms. *arXiv preprint arXiv:2601.22527*.

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*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Dataset Statistics

	XSum	CNN/DM	arXiv
Test examples	11,334	11,490	6,440
Avg src length (words)	376	684	5,906
Avg ref length (words)	21	55	163
Avg ref length (chars)	125	312	969

Table 2: Dataset statistics for the test splits used in our evaluation.

B Human Evaluation

We conduct a blind human evaluation on 50 CNN/DailyMail examples comparing DELTA+Dream, LA-SF-AR, and GPT-4o-mini. Two annotators independently rated anonymized and randomly ordered outputs on relevance, consistency, coherence, and fluency using a 1–5 Likert scale.

Method	Rel	Con	Coh	Flu	Avg.
DELTA+Dream	4.05	4.33	4.31	4.40	4.27
LA-SF-AR	4.18	4.36	4.32	4.42	4.32
GPT-4o-mini	4.26	4.32	4.26	4.42	4.31

Table 3: Blind two-annotator human evaluation on 50 CNN/DailyMail examples. Within-one-point annotator agreement is 91.5% across dimensions.

All three methods receive similar ratings across the four dimensions, providing additional evidence that DELTA's substantially tighter character-length control does not come at the cost of a substantial degradation in perceived summary quality.

C Fixed-Buffer Ablation

To isolate the effect of DELTA's dynamic buffer adjustment, we compare against fixed-buffer diffusion decoding using the same model, prompt, confidence schedule, and initialization as DELTA, but with mask insertion and removal disabled. As shown in Table 4, disabling mask insertion and removal substantially reduces character-length compliance while leaving summary quality broadly similar.

D DAEDAL Comparison

DAEDAL targets open-ended variable-length diffusion generation rather than generation toward a specified decoded character budget. We nevertheless evaluate DAEDAL with LLaDA using the same character-target prompts as the other generative methods.

The full comparison is reported in Table 5. DAEDAL produces reasonable summaries but does not reliably satisfy specified character budgets. Across all three datasets, DELTA reduces character-length error by a large margin, supporting the need for explicit decoded character-length estimation and correction.

E Runtime Analysis

Runtime results are reported in Table 6.

Method	XSum	CNN/DM	arXiv
DELTA+LLaDA	0.69 ± 0.43	8.59 ± 7.91	22.80 ± 22.98
DELTA+Dream	0.67 ± 0.39	8.59 ± 7.98	20.75 ± 21.72
LA-SF-AR	1.43 ± 0.33	2.19 ± 0.64	13.22 ± 8.04
DAEDAL (LLaDA)	2.50 ± 7.54	5.94 ± 4.22	$306.56 \pm 233.80^*$
BART-CNN	–	0.37 ± 0.06	–
Lead-char-snap	< 0.01	< 0.01	< 0.01
TextRank	< 0.01	< 0.01	< 0.01

Table 6: Wall-clock runtime in seconds per summary (mean $\pm$ standard deviation). *DAEDAL on arXiv is measured on a 100-example diagnostic subset because of its high runtime cost. GPT-4o-mini is omitted because concurrent API calls make per-sample latency not directly comparable.

F Lead Baseline Diagnostic

To distinguish our character-budgeted Lead-char-snap baseline from the conventional Lead-3 baseline, we additionally evaluate standard Lead-3 on CNN/DailyMail. Table 7 shows that Lead-3 achieves stronger extractive-summary quality but substantially violates the target character budget.

G LLM Judge Comparison

We compare the original GPT-4o-mini G-Eval scores with GPT-5-mini. The two judges differ substantially on consistency and coherence, particularly for extractive baselines. GPT-5-mini assigns substantially higher consistency scores to Lead-char-snap, which is more consistent with its extractive nature. We therefore use GPT-5-mini as the primary judge in the main paper; the complete comparison is reported in Table 8.

H Summarization Prompts

We use dataset-specific prompt templates for all generative methods (DELTA, LA-SF-AR, GPT-4o-mini, and DAEDAL). The `{target_len}` placeholder is replaced with the target character length and `{source}` with the source document.

CNN/DailyMail.

```
Summarize the following article in approximately {target_len} characters:\n\n{source}
```

arXiv.

```
Write an abstract for the following scientific paper in approximately {target_len} characters:\n\n{source}
```

XSum.

```
Write a one-sentence summary of the following article in approximately {target_len} characters:\n\n{source}
```

Dataset	Method	MAE	<10	R-1	R-L	BF1	Rel	Con	Coh	Flu
XSum	Fixed-buffer LLaDA	15.55	38.4	0.285	0.218	0.883	3.29	3.96	3.35	4.54
XSum	DELTA+LLaDA	2.3	99.6	0.286	0.217	0.882	3.33	3.96	3.38	4.53
XSum	Fixed-buffer Dream	15.74	38.1	0.334	0.259	0.892	2.81	3.81	3.09	4.46
XSum	DELTA+Dream	2.3	99.7	0.333	0.255	0.890	2.91	3.83	3.18	4.47
CNN/DM	Fixed-buffer LLaDA	27.35	23.9	0.423	0.282	0.878	4.21	4.69	3.97	4.86
CNN/DM	DELTA+LLaDA	2.4	99.6	0.363	0.259	0.876	4.19	4.66	3.94	4.84
CNN/DM	Fixed-buffer Dream	26.20	24.6	0.433	0.288	0.881	4.02	4.71	3.95	4.89
CNN/DM	DELTA+Dream	3.0	99.3	0.409	0.281	0.880	4.02	4.66	3.92	4.86
arXiv	Fixed-buffer LLaDA	113.94	4.4	0.448	0.235	0.838	4.72	4.76	4.39	4.89
arXiv	DELTA+LLaDA	2.8	98.4	0.443	0.234	0.837	4.65	4.69	4.31	4.89
arXiv	Fixed-buffer Dream	92.31	8.1	0.432	0.238	0.854	3.93	4.29	3.64	4.47
arXiv	DELTA+Dream	3.1	98.3	0.431	0.236	0.853	3.94	4.28	3.62	4.45

Table 4: Fixed-buffer ablation. Disabling DELTA's insert/remove operations substantially reduces character-length compliance while leaving summary quality broadly similar. G-Eval scores use GPT-4o-mini.

Dataset	Method	n	MAE	<10	R-1	R-L	BF1	Rel	Con	Coh	Flu	Time
XSum	DAEDAL (LLaDA)	11,334	74.5	7.7	0.251	0.196	0.845	2.82	3.41	2.81	4.25	2.50
XSum	DELTA+LLaDA	11,334	2.3	99.6	0.286	0.217	0.882	3.33	3.96	3.38	4.53	0.69
CNN/DM	DAEDAL (LLaDA)	11,490	192.3	3.4	0.367	0.248	0.858	4.00	4.43	3.67	4.69	5.94
CNN/DM	DELTA+LLaDA	11,490	2.4	99.6	0.363	0.259	0.876	3.73	4.40	3.64	4.70	8.59
arXiv	DAEDAL (LLaDA)	100	6337.5	1.0	0.106	0.073	0.796	2.64	2.92	2.34	2.90	306.56
arXiv	DELTA+LLaDA	6,440	2.8	98.4	0.443	0.234	0.837	4.65	4.69	4.31	4.89	22.80

Table 5: Comparison with DAEDAL. Time is wall-clock seconds per summary. DAEDAL on arXiv is evaluated on a 100-example diagnostic subset because of its high runtime cost; DELTA+LLaDA uses the full test set. G-Eval scores use GPT-4o-mini.

H.1 G-Eval Evaluation Prompt

We use GPT-5-mini as the primary judge with the following system and user prompts. Scores range from 1 (poor) to 5 (excellent). We additionally evaluate the same outputs with GPT-4o-mini for the judge comparison reported in Appendix G.

System prompt.

```
You are an expert evaluator tasked with assessing the quality of text summaries.

Evaluate the provided summary against the source text based on the following criteria:

- Relevance: Does the summary capture only the most important information from the source text?

- Consistency: Is the summary factually consistent with the source text?

- Coherence: Is the summary well-structured and logically organized?

- Fluency: Is the summary written in clear, grammatical English?

Score each criterion on a scale from 1 to 5:

1: Poor - Fails to meet the criterion

2: Below Average

3: Average

4: Good

5: Excellent - Perfectly meets the criterion

Respond only in the following JSON format, where X is your score:

{"relevance": X, "consistency": X, "coherence": X, "fluency": X}
```

User prompt.

```
Evaluate the following summary against the source text based on the criteria defined above.

[Source Text]: '{source}'

[Summary]: '{summary}'

Respond only in the following JSON format, where X is your integer score for each criterion:

{"relevance": X, "consistency": X, "coherence": X, "fluency": X}
```

I Implementation Details

TextRank. TF-IDF sentence scoring (Mihalcea and Tarau, 2004), truncated to the target character length at the nearest sentence boundary.

Method	Mean chars	MAE	<10	R-1	R-L	BF1	Rel	Con	Coh	Flu
Lead-char-snap	252.5	59.4	21.4	0.363	0.230	0.868	2.85	4.92	4.22	4.40
Standard Lead-3	455.4	174.8	2.6	0.400	0.250	0.869	3.51	4.96	4.67	4.75
DELTA+Dream	311.1	3.0	99.3	0.409	0.281	0.880	3.97	4.72	4.75	4.73

Table 7: Standard Lead-3 diagnostic on CNN/DailyMail. Lead-3 achieves stronger extractive-summary quality than Lead-char-snap but substantially violates the target character budget. G-Eval scores use GPT-5-mini.

Dataset	Method	Judge	Rel	Con	Coh	Flu
XSum	DELTA+LLaDA	GPT-4o-mini	3.33	3.98	3.38	4.54
		GPT-5-mini	3.41	4.21	4.61	4.66
XSum	DELTA+Dream	GPT-4o-mini	2.90	3.84	3.18	4.48
		GPT-5-mini	3.17	4.14	4.76	4.82
XSum	LA-SF-AR	GPT-4o-mini	3.49	4.02	3.55	4.64
		GPT-5-mini	3.56	4.38	4.79	4.89
XSum	GPT-4o-mini	GPT-4o-mini	3.85	4.33	3.87	4.85
		GPT-5-mini	3.83	4.73	4.96	4.99
XSum	Lead-char-snap	GPT-4o-mini	1.95	3.54	1.77	3.94
		GPT-5-mini	1.80	4.72	3.08	3.31
XSum	TextRank	GPT-4o-mini	1.70	3.69	1.69	4.15
		GPT-5-mini	1.74	4.79	4.07	4.59
CNN/DM	DELTA+LLaDA	GPT-4o-mini	4.19	4.66	3.94	4.84
		GPT-5-mini	4.04	4.59	4.58	4.49
CNN/DM	DELTA+Dream	GPT-4o-mini	4.00	4.65	3.91	4.85
		GPT-5-mini	3.97	4.72	4.75	4.73
CNN/DM	LA-SF-AR	GPT-4o-mini	4.48	4.75	4.14	4.97
		GPT-5-mini	4.29	4.61	4.94	4.96
CNN/DM	GPT-4o-mini	GPT-4o-mini	4.79	4.89	4.34	4.98
		GPT-5-mini	4.53	4.86	4.99	5.00
CNN/DM	Lead-char-snap	GPT-4o-mini	3.00	4.42	3.07	4.50
		GPT-5-mini	2.85	4.92	4.22	4.40
CNN/DM	TextRank	GPT-4o-mini	2.64	4.08	2.67	4.04
		GPT-5-mini	2.40	4.80	3.38	4.07
CNN/DM	BART-CNN	GPT-4o-mini	3.84	4.63	3.78	4.74
		GPT-5-mini	3.69	4.83	4.60	4.61
arXiv	DELTA+LLaDA	GPT-4o-mini	4.64	4.69	4.33	4.89
		GPT-5-mini	3.71	4.35	4.43	4.39
arXiv	DELTA+Dream	GPT-4o-mini	3.94	4.27	3.63	4.44
		GPT-5-mini	3.09	4.42	3.60	3.78
arXiv	LA-SF-AR	GPT-4o-mini	4.87	4.83	4.61	4.98
		GPT-5-mini	3.98	4.28	4.78	4.98
arXiv	GPT-4o-mini	GPT-4o-mini	4.98	4.96	4.91	5.00
		GPT-5-mini	4.35	4.65	4.97	5.00
arXiv	Lead-char-snap	GPT-4o-mini	3.98	4.76	3.78	4.58
		GPT-5-mini	2.22	4.99	4.11	4.27
arXiv	TextRank	GPT-4o-mini	3.23	3.88	3.00	3.91
		GPT-5-mini	2.52	4.55	3.00	3.19

Table 8: Comparison of GPT-4o-mini and GPT-5-mini as G-Eval judges.

Lead-char-snap. Selects the maximum number of initial source sentences that fit within the target character budget. This avoids unnatural mid-sentence truncation while providing an extractive baseline with approximate character-length control.

BART-CNN. BART-large fine-tuned on CNN/-DailyMail (Lewis et al., 2020), beam size 4, length penalty 2.0.

GPT-4o-mini. Prompted with the target character count using the templates in Appendix H.

LA-SF-AR. The zero-shot pipeline of Retkowski and Waibel (2025) using Llama-3.1-8B-Instruct with length approximation (μ_w=6.31 chars/word), sample filtering (N=3, temperature 0.7), and automated revisions (up to 3, ϵ=10%).

DAEDAL. We evaluate DAEDAL (Li et al., 2026) with LLaDA using the same dataset-specific character-target prompts as the other generative methods. On arXiv, we evaluate DAEDAL on a 100-example diagnostic subset because of its high inference cost.

DELTA hyperparameters. On CNN/DailyMail and XSum: $(\tau, \tau_{\text{low}}) = (0.65, 0.95)$ for Dream, $(0.65, 0.90)$ for LLaDA. On arXiv: $(0.90, 0.82)$ for Dream, $(0.71, 0.71)$ for LLaDA. Selected via grid search on the validation set. The initial buffer size is determined using the average number of characters per token estimated from validation references. The maximum number of iterations is set to $\lfloor L^*/2 \rfloor$.

Reproducibility. We release our implementation, fixed-buffer ablation code, DAEDAL evaluation script, prompts, hyperparameters, random seeds, and evaluation scripts at https://github.com/DFKI-NLP/delta-diffusion.

J Use of AI Assistants

Claude (Anthropic) was used to assist with writing and editing portions of the manuscript, generating figure components, and developing parts of the codebase. All outputs were reviewed and verified by the authors.