TRAINING-FREE LENGTH DISCOVERY FOR DIFFUSION LANGUAGE MODEL INFILLING

Hengchang Liu*
Gaoling School of Artificial Intelligence
Renmin University of China
liuhengchang@ruc.edu.cn

Zhao Yang*
Gaoling School of Artificial Intelligence
Renmin University of China
yangyz1230@ruc.edu.cn

Bing Su
Gaoling School of Artificial Intelligence
Renmin University of China
bingsu@ruc.edu.cn

ABSTRACT

Diffusion language models (DLMs) provide a bidirectional generation framework naturally suited for infilling, yet their performance is constrained by the pre-specified infilling length. In this paper, we reveal that DLMs possess an inherent ability to discover the correct infilling length. We identify two key statistical phenomena in the first-step denoising confidence: a local *Oracle Peak* that emerges near the ground-truth length and a systematic *Length Bias* that often obscures this signal. By leveraging this signal and calibrating the bias, our training-free method **CAL** (**C**alibrated **A**daptive **L**ength) enables DLMs to approximate the optimal length through an efficient search before formal decoding. Empirical evaluations demonstrate that CAL improves Pass@1 by up to 47.7% over fixed-length baselines and 40.5% over chat-based adaptive methods in code infilling, while boosting BLEU-2 and ROUGE-L by up to 8.5% and 9.9% in text infilling. These results demonstrate that CAL paves the way for robust DLM infilling without requiring any specialized training.

1 INTRODUCTION

Recently, Diffusion Language Models (DLMs) (Gong et al., 2024; Nie et al., 2025; Ye et al., 2025) have demonstrated significant potential in natural language processing. Unlike traditional Autoregressive (AR) models that rely on token-by-token causal generation, diffusion models generate sequences via iterative denoising over the entire sequence (Ho et al., 2020; Austin et al., 2021). While mainstream DLMs predominantly focus on chat tasks akin to AR models, this usage underutilizes their unique potential. With their natural bidirectional context modeling, diffusion models are inherently suited for infilling tasks (Lewis et al., 2020; Donahue et al., 2020) that require satisfying constraints from both the prefix and suffix. This capability is particularly critical for applications such as code completion (Bavarian et al., 2022; Fried et al., 2022), text infilling (Zhu et al., 2019; Lewis et al., 2020), and controllable generation (Zhou et al., 2023; Xiong et al., 2025), where the generated segment must syntactically and logically bridge the surrounding context.

Despite this theoretical promise, the actual performance of DLMs in infilling tasks often falls short of expectations (see Section 3). We find that this performance gap primarily stems from the lack of adaptive length adjustment in DLMs as quantified in Table 1. In chat-based tasks, DLMs can generate a specific `<eos>` token to terminate generation (Nie et al., 2025; Gong et al., 2024; Ye et al., 2025). In contrast, for infilling tasks, these models lack an intrinsic stopping mechanism and thus rely on a fixed, pre-specified mask length (Gong et al., 2025; Xie et al., 2025). Without a dynamic mechanism, the generation quality of DLMs is highly sensitive to the pre-specified mask length. As shown in Figure 1, an underestimated infilling length truncates the completion, whereas an over-

*Equal contribution.

estimated length causes the model to generate redundant or incorrect tokens to fill the remaining masked positions. This length constraint significantly limits DLM performance on infilling tasks.

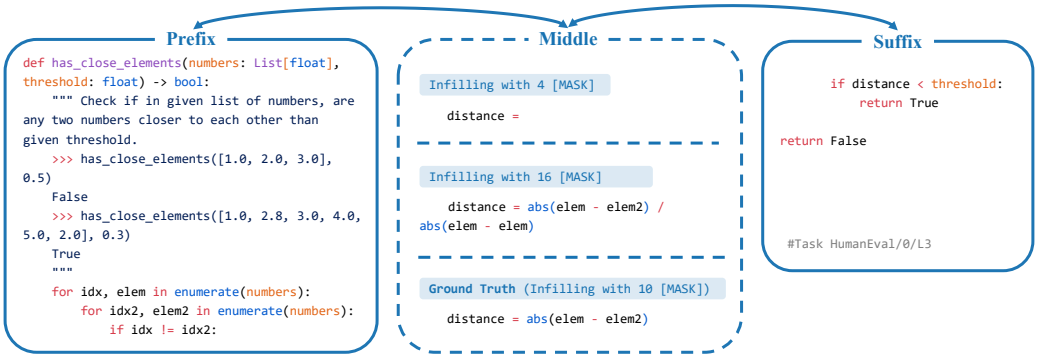


Figure 1: **Infilling sensitivity to a fixed mask length.** Evaluated on HumanEval-Infilling (Bavarian et al., 2022) using LLaDA-8B-Instruct. An underestimated length ($L = 4$) truncates the completion, whereas an overestimated length ($L = 16$) introduces redundant tokens and may yield invalid logic (e.g., division by zero) to fill the remaining masked positions. The correct solution is only recovered when the length matches the ground truth ($L = 10$).

To overcome this limitation without retraining, we investigate whether DLMs implicitly encode infilling length information within their generation process. We analyze the model's inherent confidence during first-step denoising from a fully masked state. By evaluating the average confidence across different infilling lengths, we observe two key statistical phenomena (see Figure 2): (1) *Oracle Peak*: under the same context, the average first-step confidence exhibits a local maximum when the mask length is close to the reference answer length; (2) *Length Bias*: the average confidence decreases rapidly and then gradually stabilizes as the infilling length increases.

Mechanistically, the Oracle Peak suggests that first-step confidence encodes a notion of semantic completeness: when the infilling length matches the ground-truth span, the model assigns higher certainty to its initial token predictions (see Figure 2a). In contrast, Length Bias reflects a systematic decline of average confidence as the number of masked positions increases in a fully masked state (see Figure 2b): longer masked regions weaken contextual constraints and introduce more low-certainty positions, which lowers the average token-level confidence. By calibrating the first-step confidence with the estimated bias, the length-dependent trend is largely removed, making the Oracle Peak more salient. Overall, these observations suggest that calibrated first-step confidence can serve as a practical signal for infilling length discovery, enabling adaptive length control without additional training.

Based on these findings, we propose **CAL** (**C**alibrated **A**daptive **L**ength), a training-free method for infilling length discovery. CAL introduces a length probing stage prior to formal decoding (see Figure 4 and Algorithm 1). Starting from an initial length, we evaluate the calibrated first-step confidence across a set of candidate lengths and perform a bidirectional hill-climbing search to approximate the optimal length. Since each probe requires only a single forward pass to compute the first-step logits, the additional inference overhead is modest.

Empirical evaluations on both code and text infilling benchmarks (see Table 2 and Table 3) demonstrate that CAL consistently outperforms baselines across multiple DLMs. Specifically, it improves Pass@1 by up to 47.7% over fixed-length baselines and 40.5% over chat-based adaptive methods in code infilling, while boosting BLEU-2 and ROUGE-L by up to 8.5% and 9.9% in text infilling. Our method decouples infilling performance from the rigid requirement of a pre-specified mask length, enabling more robust generation under bidirectional constraints. Our main contributions are: (1) revealing and modeling the *Oracle Peak* and *Length Bias* phenomena in first-step denoising for infilling; (2) proposing a training-free strategy for adaptive length probing via calibrated confidence; and (3) demonstrating the effectiveness of our method through extensive experiments.

2 PRELIMINARIES

Diffusion Language Models. Diffusion language models (Austin et al., 2021; Hoogeboom et al., 2021) generate sequences via iterative denoising with bidirectional context. Let $\mathbf{x}^0 = [x_1^0, \ldots, x_N^0]$ denote a clean sequence of length N. A forward noising process progressively corrupts $\mathbf{x}^0$ into $\mathbf{x}^t$ over T steps (e.g., by random masking (Sahoo et al., 2024; Shi et al., 2024)), yielding a heavily corrupted state $\mathbf{x}^T$. A DLM defines a reverse generative process parameterized by θ that reconstructs $\mathbf{x}^0$ from $\mathbf{x}^T$:

$$p_\theta(\mathbf{x}^{0:T}) = p(\mathbf{x}^T) \prod_{t=1}^{T} p_\theta(\mathbf{x}^{t-1} \mid \mathbf{x}^t). \tag{1}$$

In practice, many modern DLMs (e.g., LLaDA (Nie et al., 2025)) are trained to predict the clean tokens directly from a noisy state, i.e., to model $p_\theta(\mathbf{x}^0 \mid \mathbf{x}^t)$. The resulting token distributions provide the confidence signal used throughout our analysis and length discovery method.

Infilling Task Formulation. The infilling task (Bavarian et al., 2022; Fried et al., 2022; Zhu et al., 2019) aims to generate a semantically consistent middle segment M of length L, given a prefix context P and a suffix context S. Formally, the complete sequence is $\mathbf{x} = [P; M; S]$. While this formulation applies broadly, a critical constraint in diffusion-based infilling is that the length L must be pre-specified to initialize the masked input state $\mathbf{x}^T$. Specifically, the model constructs an initial noisy input $\mathbf{x}^T = [P; \texttt{[MASK]}^L; S]$ and denoises the masked region conditioned on the bidirectional context. Since the optimal length L^* is typically unknown a priori, a mismatched L inevitably degrades generation quality, as the model cannot dynamically adjust the span size.

First-Step Denoising Confidence. To address the unknown length challenge without auxiliary training, we leverage the intrinsic uncertainty estimates provided by the diffusion model. We focus specifically on the first-step denoising, which corresponds to the model's initial prediction $p_\theta(\mathbf{x}^0 \mid \mathbf{x}^T)$ given the fully masked input state $\mathbf{x}^T = [P; \texttt{[MASK]}^L; S]$. This prediction represents the model's "first guess" of the complete sequence based on the bidirectional context. Crucially, we isolate the probabilities corresponding to the masked segment M. Let $p_j(v)$ denote the probability assigned to token v at the j-th position of the generated sequence. We define the average first-step confidence $\Phi(L)$ over the masked indices $\mathcal{I}_{mask}$ as:

$$\Phi(L) = \frac{1}{L} \sum_{j \in \mathcal{I}_{mask}} \max_{v \in \mathcal{V}} p_j(v), \tag{2}$$

where $\mathcal{I}_{mask}$ represents the set of indices corresponding to the middle segment M, and $\mathcal{V}$ is the vocabulary. This metric serves as a computationally efficient proxy for semantic compatibility, allowing us to probe for the optimal length before committing to the full denoising trajectory.

3 ANALYSIS: LENGTH-CONFIDENCE DYNAMICS

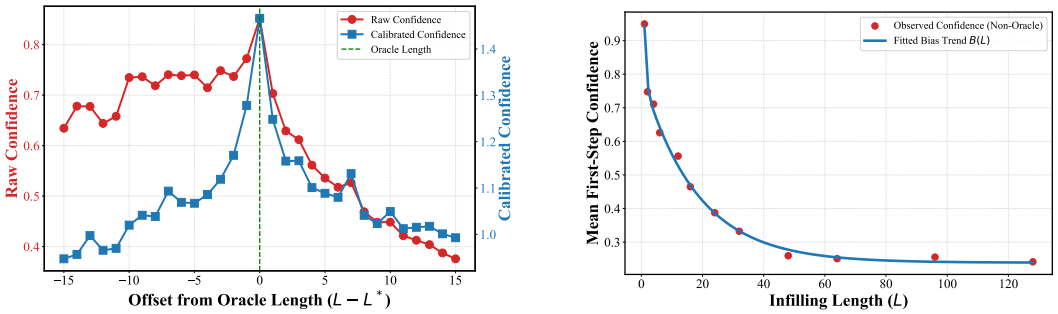
Sensitivity of Diffusion Infilling to Mask Length. Existing DLMs (Nie et al., 2025; Ye et al., 2025) typically follow a training pipeline of pre-training followed by instruction fine-tuning (SFT). During pre-training, the model learns to reconstruct text under random masks, while during SFT, it is optimized for instruction-response pairs. While effective for standard chat generation, this paradigm lacks explicit supervisory signals for `<eos>` or `<padding>` tokens within a masked span. Consequently, these models inherently lack the ability to automatically adjust infilling lengths. As illustrated in Figure 1, generation quality highly depends on the preset [MASK] length: an underestimated length forces semantic truncation, while an overestimated length causes the model to hallucinate redundant content to fill the remaining slots. Experiments confirm that providing the Oracle length (the token count of the ground-truth answer) significantly boosts performance (see Table 1), suggesting that length discovery is the primary bottleneck for unleashing the potential of diffusion models.

Oracle Peak: The Inherent Signal of Semantic Completeness. Although diffusion models lack an explicit stopping mechanism, we discover that a strong signal of logical completeness is encoded within their probability distributions. Inspired by confidence-guided adaptive length methods (Li et al., 2025a), we systematically analyze the evolution of the average first-step denoising confidence $\Phi(L - L^*)$ as a function of the infilling length L and the Oracle length L^*. We observe that, under

Table 1: **Oracle length significantly improves DLM infilling performance.** Comparison of Pass@1 scores on HumanEval-Infilling (Bavarian et al., 2022) (Single-Line) between fixed-length settings and the Oracle length setting. The Oracle Length is determined by encoding the ground-truth answer with each model's tokenizer. "Avg. Len" indicates the average Oracle length across all test cases.

Model	Fixed Length (Pass@1)				Avg.	Oracle	
	4	8	16	32		Pass@1	Avg. Len
LLaDA-Base	22.4	51.4	55.8	48.4	44.5	**87.6**	9.5
DiffuCoder-Base	24.1	51.5	59.8	47.6	45.8	**87.4**	8.6
DreamCoder-Base	24.0	55.6	63.2	54.3	47.0	**88.1**	8.6

fixed contextual constraints, the confidence curve exhibits a significant and stable local maximum near the Oracle length (see Figure 2a), which we term the *Oracle Peak*. This phenomenon reveals that diffusion models can discriminate semantic completeness at the very onset of denoising: when the allocated mask space matches the logical ground truth in terms of information entropy, the model's prediction certainty reaches a maximum. This provides a solid physical basis for extracting length information without auxiliary training.



(a) **Oracle Peak Phenomenon.** The red line represents raw confidence, while the blue line shows confidence after bias calibration.

(b) **Length Bias Fitting.** Red dots represent empirical $\Phi(L)$ samples; the blue curve shows the fitted function $B(L)$.

Figure 2: **Statistical Analysis of First-Step Denoising Confidence $\Phi(L)$.** (a) A distinct confidence peak emerges when aligned with the ground-truth length. (b) Without length information, confidence exhibits systematic decay, necessitating calibration. Statistics are derived from 100 random HumanEval-Infilling (Bavarian et al., 2022) tasks using LLaDA-8B-Base.

Quantifying Systematic Length Bias. While observing the Oracle Peak, we also identify a systematic phenomenon that interferes with the average first-step denoising confidence $\Phi(L)$, which we term *Length Bias*. Specifically, $\Phi(L)$ exhibits an overall trend of rapid initial decline followed by gradual stabilization as L increases, often causing the Oracle Peak to be obscured by background noise. This bias inherently reflects the monotonic decay of the model's prediction certainty as the search space expands under zero-information input (full mask). As shown by the red line in Figure 2a, the raw $\Phi(L - L^*)$ values on the left side of the Oracle length are significantly higher than those on the right, displaying clear asymmetry. This semantics-agnostic bias constitutes the primary obstacle to cross-length comparison and needs to be eliminated through calibration.

Calibrating Length Bias via Fitting. To address the multi-scale decay of Length Bias, we fit a double-exponential function $B(L) = ae^{-bL} + ce^{-dL} + e$ using 100 samples from the HumanEval-Infilling benchmark with LLaDA-8B-Base (see Figure 2b). Crucially, to isolate structural bias, we exclude data points near the ground-truth length during fitting, and these samples are strictly excluded from subsequent evaluations. We define the calibrated confidence $\Phi_c(L)$ as the ratio of raw confidence to the fitted bias:

$$\Phi_c(L) = \frac{\Phi(L)}{B(L)}. \tag{3}$$

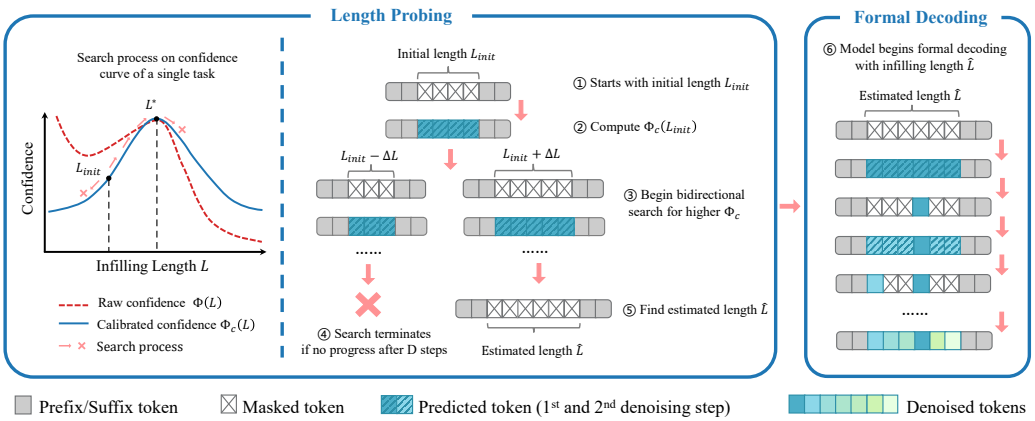


Figure 3: **Calibrated Adaptive Length Framework.** The method operates in two stages: (1) **Length Probing**, where a bidirectional hill-climbing search identifies an estimated length $\hat{L}$ to approximate the Oracle length L^* by maximizing the calibrated first-step confidence $\Phi_c(L)$; and (2) **Formal Decoding**, where the model performs standard iterative denoising initialized with the discovered length $\hat{L}$.

Mechanistically, $B(L)$ represents the expected confidence under a zero-information prior governed solely by structural constraints. Thus, $\Phi_c(L)$ functions as a signal-to-noise ratio (SNR) metric, effectively decoupling semantic signals from length-dependent statistical bias. As shown by the blue curve in Figure 2a, $\Phi_c(L - L^*)$ exhibits excellent symmetry around the Oracle length, rendering the Oracle Peak a globally prominent optimal solution. This confirms that, once structural noise is removed, the DLM's inherent confidence becomes a reliable indicator for length discovery. Specific fitting details are provided in Appendix C. We employ this single fitted $B(L)$ across all subsequent experiments, demonstrating its cross-model and cross-task robustness in Section 5.

4 METHOD

Building on the statistical findings in Section 3, we propose CAL (Calibrated Adaptive Length), a training-free framework for infilling length discovery. As illustrated in Figure 4, CAL introduces a length probing stage prior to DLM formal decoding. This stage formulates length selection as a discrete optimization problem over a calibrated confidence landscape, aiming to approximate the Oracle length to enhance infilling performance.

Optimization Objective. As demonstrated in Section 3, DLMs achieve optimal infilling performance when provided with the Oracle length, a condition often signaled by an Oracle Peak in the first-step denoising confidence. Therefore, our primary goal is to find an estimated length $\hat{L}$ that best approximates the Oracle length L^* by leveraging this intrinsic signal. However, directly maximizing the raw confidence $\Phi(L)$ is suboptimal because the Oracle Peak is typically obscured by systematic Length Bias. To address this limitation, we formulate length discovery as an optimization problem maximizing the calibrated confidence $\Phi_c(L)$. Consequently, we determine the Oracle length estimate $\hat{L}$ via the following maximization:

$$\hat{L} = \arg\max_{L \in \mathcal{R}} \Phi_c(L), \tag{4}$$

where $\mathcal{R}$ denotes the search space of candidate lengths.

Bidirectional Search Algorithm. Exhaustively computing $\Phi_c(L)$ for every candidate length is computationally inefficient. However, our analysis in Section 3 suggests that the Oracle Peak exhibits a quasi-unimodal property. Based on this observation, we employ a bidirectional hill-climbing strategy to locate the optimum efficiently. As detailed in Algorithm 1, the algorithm starts from an initial heuristic estimate L_{init} and probes the local trend of $\Phi_c(L)$ in both increasing and decreasing directions. Although noise may exist, the overall trend facilitates rapid convergence. The search terminates when the calibrated confidence fails to improve for D consecutive steps. Following

length discovery, we proceed to formal decoding by executing the full denoising process with the discovered length $\hat{L}$. Since the discovery stage requires only single-pass forward computations, it improves infilling performance with acceptable additional latency. A detailed case study illustrating this process and its robustness is provided in Appendix F.

5 EXPERIMENT

5.1 EXPERIMENTAL SETUP

We evaluate the effectiveness of CAL on both code and text infilling benchmarks.

Code Infilling. We utilize the HumanEval-Infilling benchmark (Bavarian et al., 2022), a standard dataset for Python code completion, evaluating on both Single-Line and Multi-Line tasks. Crucially, the 100 samples used for Length Bias fitting in Section 3 were strictly excluded from the test set. We apply our method to four representative discrete diffusion language models: the general-purpose LLaDA-8B (Base and Instruct) (Nie et al., 2025), and the code-specialized DiffuCoder-7B-Base (Gong et al., 2025) and DreamCoder-Base-7B (Xie et al., 2025). Additionally, we adapt DAEDAL (Li et al., 2025a), a confidence-based training-free method originally designed for chat tasks, as a baseline on LLaDA-8B-Base. All models employ greedy sampling for formal decoding, and we report Pass@1 as the primary metric.

Text Infilling. To assess generalization across diverse domains, we conduct experiments on three datasets: ROCStories (short five-sentence stories) (Mostafazadeh et al., 2016), CS Abstracts (academic writing) (Donahue et al., 2020), and Yelp Reviews (informal user-generated content) (Zhang et al., 2015). For each sample, we randomly mask a contiguous span of 2 to 8 tokens. Experiments are conducted using LLaDA-8B-Base. We employ greedy sampling and report BLEU-2 and ROUGE-L scores. For detailed hyperparameters and implementation specifics, please refer to Appendix D.

5.2 CODE INFILLING RESULTS

Table 2 presents the comprehensive results on code infilling.

Consistent Improvement over Baselines. Our adaptive length discovery method consistently achieves superior performance across all models and initial length configurations compared to fixed-length baselines. For instance, on the Single-Line task, our approach improves the average Pass@1 of LLaDA-Base from 44.8% to 65.5%. As indicated by the *Stps.* column, achieving this substantial performance gain requires only 11 to 18 additional first-step forward passes on average. This overhead is acceptable given the substantial capability improvements.

Outperforming Local-Confidence Methods. Notably, our method significantly outperforms DAEDAL, a confidence-based adaptive strategy originally designed for DLM chat tasks. While effective for open-ended generation, DAEDAL's strategy of locally inserting masks based on token-level confidence proves suboptimal for infilling. As shown in Table 2, it occasionally underperforms even the fixed-length baseline, particularly when the initial length is overestimated (e.g., $L = 16$ or 32). This performance gap highlights two key advantages of our method. First, our *bidirectional search* allows for both increasing and decreasing the span length, whereas DAEDAL is primarily limited to extending it. Second, relying on *global average confidence* proves more effective than local heuristics. Since DLMs model the joint probability of all tokens holistically, the average confidence over the entire span serves as a more reliable indicator of semantic completeness.

Performance Trends and Robustness. We observe that the performance gain diminishes as the initial length increases. We attribute this to two factors: (1) *Signal Decay.* The HumanEval-Infilling dataset is biased towards short completions (average Oracle length ≈ 9.5, see Table 1). Consequently, a large initial length (e.g., $L = 32$) implies a significant deviation from the ground truth, causing the Oracle Peak signal to decay and become obscured by background noise (as evidenced by confidence fluctuations when $|L - L^*| > 10$ in Figure 2a). (2) *Increased Prediction Difficulty.* As the masked span expands, the first-step reconstruction task becomes inherently more challenging, reducing the model's capacity to produce a sharp confidence peak. Yet, our method consistently outperforms fixed-length baselines even in these difficult scenarios, proving its robustness.

Table 2: **Code Infilling results.** We compare the Pass@1 scores of various models under four fixed-length settings ($L = 4, 8, 16, 32$) against our proposed adaptive length discovery algorithm (+ ours). Our method consistently outperforms fixed-length baselines across all initial length configurations. Stps. denotes the average number of search steps incurred during the length discovery phase.

Model	Length = 4		Length = 8		Length = 16		Length = 32		Avg.	
	Pass@1	Stps.	Pass@1	Stps.	Pass@1	Stps.	Pass@1	Stps.	Pass@1	Stps.
Single-Line Infilling										
LLaDA-Base	22.7	–	51.0	–	56.6	–	49.0	–	44.8	–
+ daedal	29.9	–	52.8	–	53.4	–	43.1	–	44.8	–
+ cal	**70.4**	11.6	**73.6**	12.3	**65.2**	14.8	**52.6**	15.7	**65.5**	13.6
LLaDA-Instruct	23.0	–	54.9	–	64.0	–	57.0	–	49.7	–
+ cal	**74.6**	12.2	**76.9**	13.0	**69.4**	14.3	**58.7**	15.3	**69.9**	13.7
DiffuCoder	24.5	–	51.5	–	60.1	–	47.7	–	46.0	–
+ cal	**73.1**	11.1	**74.8**	12.4	**68.4**	14.6	**55.8**	18.2	**68.0**	14.1
DreamCoder	24.7	–	55.4	–	63.1	–	53.9	–	49.3	–
+ cal	**75.1**	11.1	**76.2**	12.4	**70.5**	14.6	**59.0**	18.1	**70.2**	14.1
Multi-Line Infilling										
LLaDA-Base	4.6	–	10.8	–	18.7	–	27.6	–	15.4	–
+ daedal	6.0	–	12.3	–	20.4	–	26.5	–	16.3	–
+ cal	**17.1**	12.0	**21.2**	13.3	**27.1**	14.7	**31.7**	15.3	**24.3**	13.8
LLaDA-Instruct	4.7	–	11.7	–	21.4	–	30.4	–	17.0	–
+ cal	**18.8**	12.0	**22.8**	13.1	**30.2**	13.8	**34.0**	14.8	**26.5**	13.4
DiffuCoder	5.0	–	11.7	–	21.6	–	30.8	–	17.3	–
+ cal	**19.4**	12.1	**24.5**	13.4	**32.0**	14.3	**38.0**	16.6	**28.5**	14.1
DreamCoder	5.1	–	12.4	–	22.9	–	33.1	–	18.4	–
+ cal	**19.9**	12.1	**24.4**	13.3	**31.9**	14.5	**37.3**	16.9	**28.4**	14.2

5.3 Text Infilling Results

Table 3 extends our evaluation to general text domains, including short stories, scientific abstracts, and reviews.

Generalization Across Domains. Our method consistently surpasses fixed-length baselines in both BLEU-2 and ROUGE-L metrics across all datasets. For instance, on the Stories dataset, we achieve a relative improvement of 30.0% in BLEU-2 (from 15.0 to 19.5) compared to the average baseline performance. These results indicate that the discovered length signal is not limited to code syntax but reflects a fundamental semantic property of diffusion language models, proving effective for diverse natural language generation tasks.

Modality-Specific Performance Variations. We observe that the performance gains in text infilling are more modest compared to code infilling. We attribute this discrepancy to the inherent differences between the two modalities. Code is governed by strict syntax and rigid logical constraints (e.g., variable definitions), which significantly narrow the search space for the ground truth. This constraint allows the model to produce a sharp and distinct Oracle Peak. In contrast, natural language is inherently ambiguous and open-ended; the rich semantic space allows for multiple valid completions of varying lengths. This ambiguity leads to a more diffuse probability distribution, resulting in a less prominent Oracle Peak signal compared to the code domain.

5.4 Ablation Study

Necessity of Length Bias Calibration. To validate the critical role of bias calibration, we perform comprehensive comparisons across different initial lengths, models, and data modalities (see Table 4). Specifically, we utilize the HumanEval-Infilling Single-Line task for code infilling and the ROCStories dataset for text infilling. The results indicate that while the uncalibrated CAL method (w/o LB) generally outperforms the fixed-length baseline, applying Length Bias calibration yields significantly more pronounced improvements.

Table 3: **Text Infilling results.** Comparisons of BLEU-2 and ROUGE-L scores on three text datasets using LLaDA-8B-Base. The results are averaged over all test samples. Fixed-length baselines are reported for preset lengths of 2, 4, and 8 tokens. Our adaptive method consistently outperforms the baselines, demonstrating robust generalization across different textual domains.

Dataset	Model	Length = 2		Length = 4		Length = 8		Avg.	
		B-2	R-L	B-2	R-L	B-2	R-L	B-2	R-L
Stories	LLaDA-Base	9.1	25.8	18.0	37.9	17.8	37.6	15.0	33.8
	+ CAL	**17.6**	**35.7**	**19.8**	**38.4**	**21.2**	**40.7**	**19.5**	**38.3**
Abstract	LLaDA-Base	8.7	26.8	17.8	38.7	19.3	39.5	15.3	35.0
	+ CAL	**17.1**	**36.5**	**19.6**	**39.6**	**22.1**	**42.4**	**19.6**	**39.5**
Yelp	LLaDA-Base	5.9	20.1	11.4	**30.5**	12.5	30.3	9.9	27.0
	+ CAL	**10.4**	**27.8**	**11.9**	30.2	**13.6**	**32.5**	**12.0**	**30.2**

Impact at short lengths. Calibration is particularly vital for short sequences. As shown in panels (a) and (c) of Table 4, the performance gain is substantially larger when the initial length is small. This trend holds for both code and text tasks. The reason lies in the nature of Length Bias: as shown in Figure 2b, the bias curve decays most steeply at shorter lengths. Consequently, without calibration, raw confidence scores are artificially high for extremely short sequences. This often misleads the search into selecting truncated solutions. Our calibration effectively corrects this skew, allowing the model to recover the true Oracle signal.

Effective Across Models and Domains. We fit the calibration function $B(L)$ using only LLaDA-Base on a subset of HumanEval-Infilling data. Yet, this same function demonstrates excellent performance on other models (DiffuCoder, DreamCoder) and even distinct modalities (text infilling). This success across models and domains suggests that Length Bias is an inherent statistical property of DLMs, rather than a model-specific artifact. Consequently, a single fitted bias function can be robustly transferred to various diffusion models, eliminating the need for re-fitting on each new model.

Table 4: **Ablation Study of Length Bias (LB) Calibration.** We demonstrate the necessity and robustness of LB calibration across three dimensions: (a) varying initial lengths on code infilling using LLaDA-Base; (b) different model on code infilling ($L_{init} = 8$), where LLa., Dif., and Dre. correspond to LLaDA-Instruct, DiffuCoder-Base, and DreamCoder-Base, respectively; and (c) varying initial lengths on text infilling.

Method	(a) Length (Code)			(b) Models (Code)			(c) Length (Text)		
	Pass@1			Pass@1			BLEU-2 / ROUGE-L		
	$L = 4$	$L = 8$	$L = 16$	LLa.	Dif..	Dre.	$L = 2$	$L = 4$	$L = 8$
Fixed Length	22.7	51.0	56.6	54.9	51.5	55.4	9.1 / 25.8	18.0 / 37.9	17.8 / 37.6
CAL *w/o LB*	35.8	59.6	60.8	66.5	69.1	67.1	14.5 / 32.1	16.9 / 35.2	19.5 / 38.4
CAL *w/ LB*	**70.4**	**73.6**	**65.2**	**76.8**	**74.8**	**76.2**	**17.6 / 35.7**	**19.8 / 38.4**	**21.2 / 40.7**
LB Improvement	*+34.6*	*+14.0*	*+4.4*	*+10.3*	*+5.7*	*+9.1*	*+3.1 / +3.6*	*+2.9 / +3.2*	*+1.7 / +2.3*

6 Conclusion

In this paper, we address a critical limitation of Diffusion Language Models (DLMs) in infilling tasks: their sensitivity to pre-specified mask lengths. Our analysis reveals that DLMs naturally encode a length signal within their first-step denoising confidence. We characterize this through two key phenomena: the *Oracle Peak*, which signals semantic completeness, and *Length Bias*, a systematic confidence decay that must be calibrated. Leveraging these insights, we propose CAL (Calibrated Adaptive Length), a training-free framework for adaptive length discovery. By calibrating the bias and employing an efficient bidirectional search, CAL approximates the optimal infilling length prior to formal decoding. Extensive experiments on code and text benchmarks demonstrate that our approach consistently outperforms fixed-length baselines across diverse models and domains.

Limitations and Future Work. Our framework currently relies on a pre-fitted bias function. While our analysis confirms its robustness across domains, developing an online estimation method that adapts dynamically to each input represents a promising direction for future improvement. Additionally, although the search overhead is acceptable given the significant performance gains, the introduced latency might still be a consideration for ultra-low-latency real-time applications. Furthermore, our current evaluation focuses on single-span infilling; extending the calibrated search mechanism to handle multi-span or disjoint infilling scenarios remains a valuable direction for future research.

REFERENCES

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Mohammad Bavarian, Heewoo Jun, Nikolas Tezak, John Schulman, Christine McLeavey, Jerry Tworek, and Mark Chen. Efficient training of language models to fill in the middle. *arXiv preprint arXiv:2207.14255*, 2022.

Chris Donahue, Mina Lee, and Percy Liang. Enabling language models to fill in the blanks. *arXiv preprint arXiv:2005.05339*, 2020.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 320–335, 2022.

Daniel Fried, Armen Aghajanyan, Jessy Lin, Sida Wang, Eric Wallace, Freda Shi, Ruiqi Zhong, Wen-tau Yih, Luke Zettlemoyer, and Mike Lewis. Incoder: A generative model for code infilling and synthesis. *arXiv preprint arXiv:2204.05999*, 2022.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, et al. Scaling diffusion language models via adaptation from autoregressive models. *arXiv preprint arXiv:2410.17891*, 2024.

Shansan Gong, Ruixiang Zhang, Huangjie Zheng, Jiatao Gu, Navdeep Jaitly, Lingpeng Kong, and Yizhe Zhang. Diffucoder: Understanding and improving masked diffusion models for code generation. *arXiv preprint arXiv:2506.20639*, 2025.

Marton Havasi, Brian Karrer, Itai Gat, and Ricky TQ Chen. Edit flows: Flow matching with edit operations. *arXiv preprint arXiv:2506.09018*, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in neural information processing systems*, 34:12454–12465, 2021.

Bumjun Kim, Dongjae Jeon, Dueun Kim, Wonje Jeung, and Albert No. Rainbow padding: Mitigating early termination in instruction-tuned diffusion llms. *arXiv preprint arXiv:2510.03680*, 2025a.

Jaeyeon Kim, Lee Cheuk-Kit, Carles Domingo-Enrich, Yilun Du, Sham Kakade, Timothy Ngotiaoco, Sitan Chen, and Michael Albergo. Any-order flexible length masked diffusion. *arXiv preprint arXiv:2509.01025*, 2025b.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 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*, pp. 7871–7880, 2020.

Jinsong Li, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Jiaqi Wang, and Dahua Lin. Beyond fixed: Training-free variable-length denoising for diffusion large language models. *arXiv preprint arXiv:2508.00819*, 2025a.

Shufan Li, Konstantinos Kallidromitis, Hritik Bansal, Akash Gokul, Yusuke Kato, Kazuki Kozuka, Jason Kuen, Zhe Lin, Kai-Wei Chang, and Aditya Grover. Lavida: A large diffusion model for vision-language understanding. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=6WnBITpnzD.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 839–849, 2016.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, JUN ZHOU, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=KnqiC0znVF.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. Insertion transformer: Flexible sequence generation via insertion operations. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5976–5985. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/stern19a.html.

Zirui Wu, Lin Zheng, Zhihui Xie, Jiacheng Ye, Jiahui Gao, Yansong Feng, Zhenguo Li, Victoria W., Guorui Zhou, and Lingpeng Kong. Dreamon: Diffusion language models for code infilling beyond fixed-size canvas, 2025. URL https://hkunlp.github.io/blog/2025/dreamon.

Zhihui Xie, Jiacheng Ye, Lin Zheng, Jiahui Gao, Jingwei Dong, Zirui Wu, Xueliang Zhao, Shansan Gong, Xin Jiang, Zhenguo Li, et al. Dream-coder 7b: An open diffusion language model for code. *arXiv preprint arXiv:2509.01142*, 2025.

Zhen Xiong, Yujun Cai, Zhecheng Li, and Yiwei Wang. Unveiling the potential of diffusion large language model in controllable generation. *arXiv preprint arXiv:2507.04504*, 2025.

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

Zebin You, Shen Nie, Xiaolu Zhang, Jun Hu, Jun Zhou, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. Llada-v: Large language diffusion models with visual instruction tuning. *arXiv preprint arXiv:2505.16933*, 2025.

Andrew Zhang, Anushka Sivakumar, Chia-Wei Tang, and Chris Thomas. Flexible-length text infilling for discrete diffusion models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 31332–31347, 2025.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. A reparameterized discrete diffusion model for text generation. *arXiv preprint arXiv:2302.05737*, 2023.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Ethan Wilcox, Ryan Cotterell, and Mrinmaya Sachan. Controlled text generation with natural language instructions. In *International Conference on Machine Learning*, pp. 42602–42613. PMLR, 2023.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, et al. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. *arXiv preprint arXiv:2505.19223*, 2025.

Wanrong Zhu, Zhiting Hu, and Eric Xing. Text infilling. *arXiv preprint arXiv:1901.00158*, 2019.

A Related Work

Diffusion Language Models. Discrete diffusion models have recently emerged as a compelling alternative to auto-regressive models, enabling probabilistic modeling of token sequences through iterative denoising. Foundational works (Austin et al., 2021; Hoogeboom et al., 2021) introduced discrete corruption processes, which were further refined by improved training objectives (Zheng et al., 2023; Lou et al., 2023). Within this domain, Masked Diffusion Models (MDMs) have become a dominant paradigm, utilizing masking kernels to achieve scalable training (Sahoo et al., 2024; Shi et al., 2024). Recent efforts have pushed these models to the billion-parameter frontier: LLaDA (Nie et al., 2025) pioneered large-scale training from scratch, while its successor LLaDA-1.5 (Zhu et al., 2025) has integrated reinforcement learning for alignment, and LLaDA-V (You et al., 2025) has extended to multimodal understanding. Parallel research, such as DiffuLLaMA (Gong et al., 2024) and Dream (Ye et al., 2025), explored adapting pre-trained AR models into diffusion frameworks. Furthermore, specialized models like DiffuCoder (Gong et al., 2025) and DreamCoder (Xie et al., 2025) have been developed specifically for code generation. While leveraging the bidirectional nature of diffusion, most prior work has focused primarily on standard instruction-following capabilities.

Generative Infilling. Early approaches to generative infilling predominantly relied on encoder-decoder architectures, such as BART (Lewis et al., 2020), T5 (Raffel et al., 2020), and Insertion Transformers (Stern et al., 2019), which treat infilling as a span corruption and reconstruction task. With the dominance of decoder-only models, research shifted toward adapting AR architectures for infilling. Techniques like Fill-In-the-Middle (FIM) (Bavarian et al., 2022; Fried et al., 2022) and blank-infilling objectives (Donahue et al., 2020; Du et al., 2022) were introduced to enable bidirectional context awareness within a causal generation framework. More recently, the rise of diffusion language models (Nie et al., 2025; Gong et al., 2024; Ye et al., 2025) has offered a new paradigm. Thanks to their non-causal nature, these models naturally model the joint distribution of the prefix, middle, and suffix. While standard DLMs typically require fixed lengths, specialized variants such as DDOT (Zhang et al., 2025), FlexMDM (Kim et al., 2025b), and others (Wu et al., 2025; Xiong et al., 2025) have been proposed to enable variable-length generation capabilities.

Adaptive Length Control in DLMs. Standard diffusion language models typically handle length variations in instruction-following tasks by initializing an oversized canvas and filling unused positions with `<eos>` or padding tokens (Nie et al., 2025; Gong et al., 2024; Ye et al., 2025). While recent works like Rainbow Padding (Kim et al., 2025a) have optimized this padding mechanism, these strategies are primarily tailored for chat-style interactions rather than infilling. To enable flexible-length generation for infilling, several training-based approaches have been proposed. For instance, Zhang et al. (2025) achieved flexible-length infilling via optimal transport, while Havasi et al. (2025) and Kim et al. (2025b) abandoned the fixed-length mask-decode paradigm in favor of operations like insertion and deletion. Similarly, Wu et al. (2025) introduced special tokens `<expand>` and `<delete>` to dynamically adjust sequence length, while LaViDa (Li et al., 2025b) relied on a specialized training stage with FIM objectives to achieve variable-length infilling. However, these methods necessitate extensive pre-training or task-specific fine-tuning.

In contrast, training-free approaches such as DAEDAL (Li et al., 2025a) attempted to dynamically adjust length at inference time in chat tasks by monitoring local token confidence. Similarly, Xiong

et al. (2025) prompted DLMs to generate <null> tokens as placeholders to regulate length in controllable JSON generation. However, these methods rely on explicit termination signals or rigid structural constraints, which are often absent in general infilling scenarios. Our work bridges this gap by proposing a training-free framework that leverages the global first-step confidence of the entire masked region. By calibrating the systematic Length Bias, we approximate the optimal infilling length without relying on additional training.

B METHOD ALGORITHM

Algorithm 1 Adaptive Length Discovery

Input: Context P, S, Model $\mathcal{M}$, Bias $B(L)$
Hyperparams: Step ΔL, Tolerance D, Init L_{init}, Max L_{max}
$\hat{L} \leftarrow L_{init}, \quad \hat{\Phi}_c \leftarrow \Phi(L_{init})/B(L_{init})$
for $dir \in \{+1, -1\}$ **do**
  $L \leftarrow L_{init} + dir \cdot \Delta L, \quad cnt \leftarrow 0$
  while $L \in [1, L_{max}] \wedge cnt < D$ **do**
    $\Phi_c \leftarrow \Phi(L)/B(L)$
    if $\Phi_c > \hat{\Phi}_c$ **then**
      $\hat{L}, \hat{\Phi}_c, cnt \leftarrow L, \Phi_c, 0$ {Update best length}
    else
      $cnt \leftarrow cnt + 1$
    end if
    $L \leftarrow L + dir \cdot \Delta L$
  end while
end for
return Denoise$(\mathcal{M}, P, S, \hat{L})$

C LENGTH BIAS FITTING DETAILS

Data Sampling and Oracle Exclusion Strategy. We randomly sampled 100 single-line infilling tasks with varying ground-truth lengths from the HumanEval-Infilling benchmark (Bavarian et al., 2022). Crucially, these tasks were strictly excluded from the test set used in our main experiments to prevent any data leakage. For each task i, we fixed the prefix P_i and suffix S_i, probing the first-step denoising confidence across a wide range of lengths $\mathcal{L}_{test} = \{1, 2, 4, 6, 12, 16, 24, 32, 48, 64, 96, 128\}$. To decouple the systematic Length Bias from semantic signals, we implemented a strict *Oracle exclusion strategy*: any probe length L falling within the neighborhood of the Oracle length L_i^* (specifically, $L \in [L_i^* - 4, L_i^* + 4]$) was discarded. This ensures that the fitted curve $B(L)$ captures the background entropy decay rather than the model's semantic confidence.

Double-Exponential Fitting. Based on the characteristic trend of *rapid initial decay followed by stabilization* (visualized in Figure 2b), we adopted a double-exponential model $B(L) = ae^{-bL} + ce^{-dL} + e$. The first exponential term accounts for the sharp dilution of local constraints in short sequences, while the second term models the long-tail asymptotic behavior. We solved for the parameters using non-linear least squares, weighting each length point L by $w_L = 1/\sqrt{N_L}$ to account for sample size variations.

Fitted Parameters and Robustness. For the LLaDA-8B-Base model (Nie et al., 2025), the optimal parameters were determined as: $a = 1.00, b = 1.77$ (fast decay), $c = 0.56, d = 0.06$ (slow decay), and $e = 0.24$ (baseline). Although fitting on different subsets may yield minor parameter fluctuations, our ablation studies (see Table 4) demonstrate that the derived bias function is highly robust. It generalizes effectively across diverse models (DiffuCoder (Gong et al., 2025), DreamCoder (Xie et al., 2025)) and domains (text infilling (Mostafazadeh et al., 2016; Donahue et al., 2020)), confirming that Length Bias is a universal statistical property of diffusion masking. We provide the full fitting script in our supplementary material to ensure reproducibility.

D EXPERIMENTAL SETUP DETAILS

Hyperparameters and Resources. For our adaptive length discovery method (CAL), we set the search step size $\Delta L = 1$ for all tasks to ensure fine-grained length probing. The search tolerance is set to $D = 4$ for code infilling tasks and $D = 2$ for text infilling tasks. Regarding the initial length configuration, we evaluated four settings for code infilling ($L_{init} \in \{4, 8, 16, 32\}$) and three settings for text infilling ($L_{init} \in \{2, 4, 8\}$). All experiments were conducted on a node equipped with 8 NVIDIA A40 (40GB) GPUs.

Input Format. We adopted the standard infilling paradigm where the input sequence is constructed as [Prefix] [MASK] [Suffix]. During the denoising process, the tokens in the prefix and suffix regions remain clamped (i.e., unmasked and fixed), while the model performs length adjustment and decoding exclusively on the masked span.

Baseline DAEDAL Implementation. We adapted DAEDAL (Li et al., 2025a), a training-free dynamic length method originally for chat tasks, to serve as our infilling baseline. The original DAEDAL algorithm consists of two stages: (1) an initial length extension based on <eos> confidence, and (2) a runtime expansion based on low-confidence tokens during decoding. For the infilling adaptation, we omitted the first stage because DLMs in infilling mode typically do not generate explicit <eos> tokens, rendering the stop-signal-based extension ineffective. We focused on the second stage, where the span length is dynamically increased if tokens exhibit low confidence. Based on tuning results (see Table 5), we adjusted the confidence thresholds to better suit the infilling distribution: the low-confidence threshold was raised to 0.3 (compared to 0.1 in the original paper), as we found 0.1 to be too conservative for infilling scenarios. The high-confidence threshold was maintained at the original value of 0.9. The expansion span was set to 2 tokens to facilitate fine-grained length adjustment. The maximum generation length was constrained to 64 for Single-Line tasks and 128 for Multi-Line tasks.

E PARAMETER ABLATION

E.1 DAEDAL THRESHOLD SENSITIVITY

We first examine the impact of the low-confidence threshold in the DAEDAL baseline. As shown in Table 5, the performance of DAEDAL is highly sensitive to this threshold. A lower threshold (e.g., 0.1 or 0.2) makes the model conservative in expanding the length, failing to correct underestimated lengths (e.g., $L = 4$). Conversely, a higher threshold (e.g., 0.5) encourages aggressive expansion, which benefits short initial lengths but degrades performance when the initial length is already sufficient ($L = 16$ or 32) by introducing unnecessary redundancy. We selected a threshold of 0.4 as the optimal trade-off, achieving the highest average Pass@1 (44.8%) across all settings, which matches the fixed-length baseline on average but provides better robustness for shorter initializations.

Table 5: **Impact of DAEDAL low-confidence threshold on code infilling performance.** Evaluations on HumanEval-Infilling (SingleLine) by LLaDA-8B-Base.

THRESHOLD	INITIAL LENGTH (PASS@1)				AVG.
	$L = 4$	$L = 8$	$L = 16$	$L = 32$	
FIXED LENGTH	22.7	51.0	**56.6**	**49.0**	**44.8**
0.1	22.7	49.8	54.8	46.0	43.3
0.2	23.6	50.0	54.8	43.5	43.0
0.3	26.4	51.2	55.6	43.5	44.2
0.4	29.9	52.8	53.4	43.1	**44.8**
0.5	**32.9**	**54.2**	50.0	38.8	44.0

E.2 CAL SEARCH PARAMETERS

We further investigate the robustness of our CAL method with respect to its two key search parameters: the tolerance D (stopping criterion) and the step ΔL (search granularity).

Impact on Code Infilling. Table 6 presents the results on the HumanEval-Infilling benchmark. We observe a clear trade-off between search accuracy and computational cost. Increasing the tolerance D from 2 to 4 leads to a performance gain (from 71.4% to 74.4%), as it prevents the search from terminating prematurely at local optima. However, this comes at the cost of increased search steps. Beyond $D = 4$, the performance plateaus while the cost continues to rise. Regarding step size, a finer granularity ($\Delta L = 1$) outperforms a coarser one ($\Delta L = 2$), confirming that precise length estimation is critical for code generation. Consequently, we adopt $D = 4$ and $\Delta L = 1$ as the default configuration for code tasks.

Impact on Text Infilling. For text infilling tasks, as shown in Table 7, the method exhibits greater robustness to parameter variations. Even with a minimal tolerance of $D = 1$, CAL significantly outperforms the fixed-length baseline. Increasing D to 2 or 3 yields marginal improvements in BLEU-2 and ROUGE-L scores. This suggests that the confidence landscape in natural language is smoother than in code, making the Oracle Peak easier to locate. To balance performance and efficiency, we select $D = 2$ and $\Delta L = 1$ for all text infilling experiments.

Table 6: **Impact of CAL search parameters on code infilling.** Evaluations on HumanEval-Infilling (SingleLine) by LLaDA-8B-Base with $L_{init} = 8$. Stps. denotes the average number of search steps incurred during the length discovery phase.

Tolerance D	Step ΔL	Pass@1	Stps.
Fixed Length ($L = 8$)		51.4	–
2	1	71.4	7.3
2	2	65.9	6.4
3	1	73.1	9.8
4	1	**74.4**	12.2
5	1	**74.4**	14.4

Table 7: **Impact of CAL parameters on text infilling performance.** Evaluations are conducted on ROCStories dataset by LLaDA-8B-Base.

Tolerance D	Step ΔL	Length = 2		Length = 4		Length = 8		Avg.	
		B-2	R-L	B-2	R-L	B-2	R-L	B-2	R-L
Fixed Length		9.1	25.8	18.0	37.9	17.8	37.6	15.0	33.8
1	1	15.0	32.6	20.2	39.5	21.0	40.9	18.7	37.7
1	2	18.0	36.4	19.5	38.3	20.7	40.2	19.4	**38.3**
2	1	17.6	35.7	19.8	38.4	21.2	40.7	19.5	**38.3**
3	1	18.8	37.1	19.5	37.9	20.8	39.9	**19.7**	**38.3**

F Case Studies

To provide a concrete understanding of how CAL operates, we present a detailed case study on the task HumanEval/0/L3 (Bavarian et al., 2022). As shown in Table 8, this task requires the model to infill the core logic for calculating the distance between two numbers within a nested loop. The ground-truth middle segment is `distance = abs(elem - elem2)\n`, which consists of exactly 10 tokens using the LLaDA tokenizer.

Search Process Analysis. Table 9 demonstrates the robustness of our bidirectional search. Whether starting from an underestimated length ($L = 4$, Table a) or an overestimated one ($L = 16$, Table b), the calibrated confidence $\Phi_c(L)$ effectively suppresses length bias and guides the algorithm to find at the Oracle length ($L = 10$).

Visualization of the Oracle Peak. Figure 4 plots the confidence landscape for this task. It clearly shows that while the raw confidence (red) decays due to structural noise (blue dashed), the calibrated score (blue solid) reveals a prominent Oracle Peak at $L = 10$. Although local fluctuations exist, the calibrated confidence metric $\Phi_c(L)$ reaches its maximum at the Oracle length.

Table 8: **Illustration of an Infilling Task Structure.** This example (HumanEval/0/L3) demonstrates the decomposition of a code generation task into a prefix (P), a ground-truth middle (M^*), and a suffix (S). The model's goal is to recover the core semantic logic within the masked span.

Prefix (P)

```
from typing import List

def has_close_elements(numbers, threshold):
  """ Check if in given list of numbers, are any
  two numbers closer to each other than given
  threshold.
  >>> has_close_elements([1.0, 2.0, 3.0], 0.5)
  False
  >>> has_close_elements([1.0, 2.8, 3.0, 4.0, 5.0,
2.0], 0.3)
  True
  """
  for idx, elem in enumerate(numbers):
    for idx2, elem2 in enumerate(numbers):
      if idx != idx2:
```

Middle (M^*)

```
        distance = abs(elem - elem2)
```

Suffix (S)

```
      if distance < threshold:
        return True

  return False
```

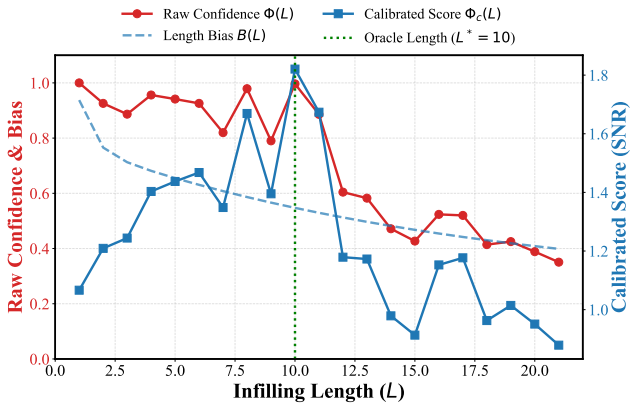


Figure 4: **Dynamics of the confidence landscape during bidirectional search.** We compare the raw confidence $\Phi(L)$ (red) and the calibrated score $\Phi_c(L)$ (blue solid). While the raw signal is biased towards shorter lengths, our calibration against the systematic bias $B(L)$ (blue dashed) reveals a prominent Oracle Peak at the ground-truth length $L^* = 10$ (green dotted line).

Table 9: **Case Studies of Bidirectional Length Probing.** We illustrate the search process for Task HumanEval/0/L3 on LLaDA-8B-Instruct (matching the example in Figure 1) starting from different initial lengths. $\Phi(L)$ denotes the raw confidence, $B(L)$ the fitted bias, and $\Phi_c(L)$ the calibrated confidence. In both scenarios, the algorithm accurately identifies the Oracle length $L^* = 10$ (highlighted), where the calibrated confidence $\Phi_c(L)$ is maximized and the model recovers the ground-truth logic.

(a) Starting from an underestimated length ($L_{init} = 4$).

Probe L	$\Phi(L)$	$B(L)$	$\Phi_c(L)$	**First-Step Denoising Prediction**
$L = 4$	0.956	0.681	1.403	`distance =\n`
$L = 5$	0.941	0.655	1.437	`distance = elem\n`
$L = 6$	0.926	0.631	1.468	`distance = elem2\n`
$L = 7$	0.820	0.608	1.348	`distance = abs(elem2\n`
$L = 8$	0.979	0.587	1.669	`distance = elem - elem2\n`
$L = 9$	0.790	0.566	1.395	`distance = abs(elem elem2)\n`
$L = 10$	0.997	0.547	**1.821**	`distance = abs(elem - elem2)\n`
$L = 11$	0.886	0.529	1.674	`distance = abs(elem - elem2)\n\n`
$L = 12$	0.604	0.513	1.179	`distance = abs(elem abs -2 elemabs)\n`
$L = 13$	0.583	0.497	1.173	`distance = abs(elem - abs(elem - elem2)\n`
$L = 14$	0.472	0.482	0.979	`distance = abs distance abs abs(elem - elem2)\n`
$L = 3$	0.887	0.713	1.244	`distance\n`
$L = 2$	0.926	0.766	1.209	`\n`
$L = 1$	1.000	0.938	1.066	`\n`

(b) Starting from an overestimated length ($L_{init} = 16$).

Probe L	$\Phi(L)$	$B(L)$	$\Phi_c(L)$	**First-Step Denoising Prediction**
$L = 16$	0.524	0.454	1.153	`distance = abs(elem - elem2) abs(elem - elem2)\n`
$L = 17$	0.520	0.442	1.177	`distance = abs(elem - elem2)\n abs(elem - elem2)\n`
$L = 18$	0.414	0.430	0.963	`distance = abs(elem - elem2) distance abs abs(elem - elem2)\n`
$L = 19$	0.425	0.419	1.014	`distance = abs(elem - elem2\n distance = abs(elem - elem2)\n`
$L = 20$	0.389	0.409	0.951	`distance = abs(elem - elem2\n\n distance = abs(elem - elem2)\n`
$L = 21$	0.351	0.399	0.879	`distance = abs(elem - elem2\n\n distance =(elem abs(elem - elem2)\n`
$L = 15$	0.427	0.468	0.913	`distance = abs(elem - -2) abs - elem2)\n`
$L = 14$	0.472	0.482	0.979	`distance = abs distance abs abs(elem - elem2)\n`
$L = 13$	0.583	0.497	1.173	`distance = abs(elem - abs(elem - elem2)\n`
$L = 12$	0.604	0.513	1.179	`distance = abs(elem abs -2 elemabs)\n`
$L = 11$	0.886	0.529	1.674	`distance = abs(elem - elem2)\n\n`
$L = 10$	0.997	0.547	**1.821**	`distance = abs(elem - elem2)\n`
$L = 9$	0.790	0.566	1.395	`distance = abs(elem elem2)\n`
$L = 8$	0.979	0.587	1.669	`distance = elem - elem2\n`
$L = 7$	0.820	0.608	1.348	`distance = abs(elem2\n`
$L = 6$	0.926	0.631	1.468	`distance = elem2\n`